

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Une approche transformationnelle pour la conception de tableurs multi-dimensionnels pour l'usage domestique

Lequeux, Jean-Yves

Award date:
2015

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Une approche transformationnelle pour la conception de tableurs multidimensionnels pour l'usage domestique

Jean-Yves Lequeux



Promoteur : Petit Michaël

Co-Promoteur : Hainaut Jean-Luc

Mémoire présenté en vue de l'obtention du grade de
Master en Sciences Informatiques

Je souhaite remercier tout particulièrement mon promoteur, monsieur Michaël Petit, qui m'a suivi et m'a conseillé tout au long de la réalisation de mon mémoire. Je remercie également toutes les personnes du corps académique de l'université de Namur qui m'ont permises de mener à bien mon cursus universitaire. Enfin, je tiens à remercier ma famille, notamment ma conjointe Virginie, et mes amis qui m'ont apporté leur aide et m'ont soutenu tout au long de ces années d'étude.

RÉSUMÉ

Cette étude propose des règles de traduction d'un problème de calcul afin d'aider un utilisateur privé ou au sein d'une petite structure professionnelle à implémenter une solution à ce problème dans un tableur. L'approche proposée consiste en deux phases de traduction. La première phase est la transformation du modèle conceptuel en modèle logique qui est ensuite lui-même traduit dans la deuxième phase en modèle physique.

MOTS-CLÉS

Modèle de calcul, tableur, grandeurs multidimensionnelles, règle de calcul, règles de traduction, modèle conceptuel, modèle physique, modèle logique, méta-modèle.

ABSTRACT

This study presents translation rules of a computing problem in order to help a private user or a person in a very small professional structure to implement a solution to this problem in a spreadsheet. The suggested approach consists in 2 translation phases. The first phase transforms the conceptual model into a logical one and this one is then translated into a physical one during the second phase.

KEYWORDS

Computing models, spreadsheet, multidimensional measures, computing rules, translation rules, conceptual model, physical model, logical model, metamodel.

Table des matières

RÉSUMÉ.....	i
MOTS-CLÉS.....	i
ABSTRACT.....	i
KEYWORDS.....	i
Introduction.....	1
1 Résolution de problème de calcul dimensionné : état de l'art	2
1.1 Solutions logicielles grand public	2
1.1.1 Microsoft® Excel.....	3
1.1.2 Autres logiciels grand public.....	9
1.2 Solutions logicielles avancées	9
1.3 Solutions méthodologiques : modèles de conception	10
2 Périmètre et objectifs du mémoire	11
3 Un problème de calcul.....	11
3.1 Étude de cas	11
3.1.1 Gestion de projet :.....	11
4 Point de départ : un modèle de calcul conceptuel.....	12
4.1 Les concepts	13
4.2 Un outil d'analyse : le graphe des dépendances.....	17
4.3 La méthodologie.....	18
4.4 Vue d'ensemble.....	18
4.5 Application concrète	20
5 Concrétisation dans un tableur : un modèle physique.....	23
5.1 Les concepts	24
5.2 Vue d'ensemble.....	24
6 Du modèle de calcul conceptuel aux feuilles de calculs.....	27
6.1 Élément intermédiaire : un modèle logique	28
6.2 Vue d'ensemble.....	32
6.3 Règles de traduction.....	34
6.3.1 Du modèle conceptuel au modèle logique	34
6.3.2 Du modèle logique au modèle physique.....	36

Conclusion	37
Bibliographie.....	38
Annexes	39
RÉSUMÉ.....	i
MOTS-CLÉS	i
ABSTRACT	i
KEYWORDS.....	i
Introduction.....	1
1 Résolution de problème de calcul dimensionné : état de l'art	2
1.1 Solutions logicielles grand public	2
1.1.1 Microsoft® Excel	3
1.1.2 Autres logiciels grand public.....	9
1.2 Solutions logicielles avancées	9
1.3 Solutions méthodologiques : modèles de conception	10
2 Périmètre et objectifs du mémoire	11
3 Un problème de calcul.....	11
3.1 Étude de cas	11
3.1.1 Gestion de projet :.....	11
4 Point de départ : un modèle de calcul conceptuel.....	12
4.1 Les concepts	13
4.2 Un outil d'analyse : le graphe des dépendances.....	17
4.3 La méthodologie.....	18
4.4 Vue d'ensemble.....	18
4.5 Application concrète	20
5 Concrétisation dans un tableur : un modèle physique.....	23
5.1 Les concepts	24
5.2 Vue d'ensemble.....	24
6 Du modèle de calcul conceptuel aux feuilles de calculs.....	27
6.1 Élément intermédiaire : un modèle logique	28
6.2 Vue d'ensemble.....	32
6.3 Règles de traduction.....	34
6.3.1 Du modèle conceptuel au modèle logique	34

6.3.2 Du modèle logique au modèle physique.....	36
Conclusion	37
Bibliographie.....	38
Annexes	39

Introduction

Dans le monde de l'entreprise mais également chez les particuliers, il est courant de rencontrer des situations qui requièrent une solution informatique pour résoudre un problème. Par exemple, le suivi d'un budget mensuel, la gestion d'un classement des joueurs d'un club de sport, l'aide à la décision pour le choix d'un investissement. Il arrive également que les solutions proposées soient trop coûteuses et/ou que le délai pour en disposer soit trop conséquent. Dans ce genre de cas, il n'est pas rare que les utilisateurs se tournent alors vers des programmes de type tableur tels que Microsoft® Excel ou LibreOffice. Bien que l'utilisation d'un tableur pour résoudre des problèmes de calcul de ce genre soit tout à fait pertinente, la tâche n'est pas toujours des plus aisées. D'autre part, l'implantation de la solution dans un tableur ou même l'évolution des besoins peuvent amener à des problèmes et des erreurs de calcul.

Dans ce mémoire, nous allons proposer une solution permettant de faciliter la mise en place de ce genre de solution. Nous recommanderons l'utilisation de modèles conceptuels dans la construction d'une solution à un modèle de calcul afin d'assurer une meilleure compréhension du problème et une meilleure définition de la solution.

Nous envisagerons ainsi une alternative de construction de solution de calcul basée sur différents niveaux d'abstraction et de modèle (conceptuel, logique et physique). Nous proposerons un ensemble de règles de traduction permettant une transition entre chaque couche d'abstraction afin de faciliter le passage du modèle conceptuel à l'implantation de la solution dans le tableur.

Nous nous intéresserons également à des problèmes qualifiés de dimensionnés. Il s'agit de problèmes dont certains aspects se définissent d'une seule manière mais qui s'expriment différemment en fonction des valeurs du critère auquel ils se rapportent.

Prenons l'exemple de la gestion d'une carte de fidélité dans un magasin quelconque. Le principe est que le client accumule des points à chaque achat qu'il réalise. Une fois qu'un certain seuil de points est atteint, le client dispose d'une réduction sur son prochain achat. Dans ce problème de calcul, on dira que les différents concepts sont dimensionnés puisqu'ils s'expriment en fonction du critère « dimension » client. En effet, la gestion de l'accumulation des points, par exemple, reste identique pour chaque client mais le résultat, c'est-à-dire le nombre de points accumulés, sera bien différent d'un client à l'autre. En revanche, si l'on souhaite connaître le client le plus fidèle, le résultat de ce calcul s'exprimera en dehors de toute dimension puisqu'il ne concerne qu'un seul client, le client ayant réalisé le plus d'achat.

1 Résolution de problème de calcul dimensionné : état de l'art

Avant de rentrer dans le vif du sujet, nous allons commencer par nous intéresser à quelques solutions déjà existantes ou en cours de développement et qui répondent de manière plus ou moins adéquate au problème qui nous intéresse dans ce document.

Nous allons principalement aborder des solutions logicielles grand public car elles ciblent plus particulièrement les utilisateurs concernés par notre problème. Nous n'envisagerons pas ici des catégories de logiciels plus poussés, qui semblent aborder les aspects multidimensionnels de manière plus complète mais qui, en contrepartie, sont moins accessibles car plus techniques. Ces solutions sont donc souvent destinées au monde de l'entreprise plutôt qu'aux utilisateurs non professionnels et concernent des problèmes beaucoup plus complexes que ceux dont nous traitons ici. Nous avons donc considéré qu'elles sortaient du champ d'étude de ce mémoire.

Plus loin dans cette étude (au point 4), nous évoquerons une des méthodes de conception existantes qui facilitent et améliorent la définition de solutions pouvant être mises en œuvre dans les logiciels précédemment introduits afin de résoudre des problèmes de calcul.

1.1 Solutions logicielles grand public

Lorsqu'il s'agit de résoudre des problèmes de calcul peu ou moyennement complexes, tels que le calcul d'un budget personnel ou l'organisation d'un projet dans une petite entreprise, les utilisateurs ont à leur disposition des logiciels appelés « tableurs ». Ces derniers fonctionnent sur le principe des feuilles de calcul où les données sont présentées en lignes et/ou en colonnes. Ces tableurs permettent ainsi aux utilisateurs de définir facilement les données initiales et les données « résultat » en précisant simplement les formules de calcul de ces dernières.

Parmi les logiciels grand public présents sur le marché, nous retrouvons entre autres Microsoft® Excel, Apache OpenOffice Calc, LibreOffice Calc, mais également des solutions en ligne telles que Google Spreadsheet¹ et Framacalc². Bien qu'ils partagent tous les mêmes fonctionnalités de base, ils se différencient généralement par les outils construits autour du tableur comme le nombre de fonctions de calcul proposé, les fonctionnalités de partage de document en ligne ou de travail collaboratif ou encore les outils facilitant le calcul ou la visualisation des données dans le tableau. Parmi ces fonctionnalités, il y en a une d'autant plus intéressante qu'elle permet la visualisation des données multidimensionnelles. Il s'agit de ce que l'on appelle les tableaux croisés dynamiques ou « table pivot ». Si l'on reprend la définition d'une table pivot selon Microsoft®, il s'agit *« d'un outil interactif qui associe et compare des données. Il permet à l'utilisateur de faire pivoter ses lignes et ses colonnes pour voir différents résumés des données source et afficher les détails qui l'intéressent... »* Cet outil permet également d'obtenir une vue d'ensemble d'un grand nombre de données sous différents angles, qu'on appellera « dimensions » dans cet exposé. Nous retrouvons également cette même fonctionnalité dans « LibreOffice Calc » sous l'appellation de « table de pilote » décrite comme

¹ <https://docs.google.com/spreadsheets/>

² <http://www.framacalc.org>

« une table interactive où les données peuvent être organisées et réorganisées ou résumées en fonction de différents points de vue. »

L'avantage de ce type de logiciels est de permettre à l'utilisateur final d'obtenir la solution de son problème par l'utilisation de formules décrivant ce même problème. Cet avantage peut rapidement se transformer en inconvénient lorsque le problème est mal décrit. Cela peut, par ailleurs, se compliquer lorsqu'il s'agit d'aborder des problèmes plus complexes. En effet, ces logiciels sont fournis avec de plus en plus de fonctionnalités. La plupart de celles-ci sont cependant rarement utilisées et souvent mal comprises par les utilisateurs qui finissent par se perdre dans les méandres du logiciel. Dans le cas où l'utilisateur utiliserait une de ces fonctionnalités moins basiques, il ne détient pas toujours le savoir et les compétences en informatique nécessaires à la bonne définition du problème, ce qui ne garantit dès lors pas l'obtention du résultat recherché. Pour pallier à ces lacunes, il est intéressant et utile de faire appel à un modèle de calcul comme nous pourrions le voir ci-après.

Si l'on s'attarde plus spécifiquement à la résolution des problèmes multidimensionnels, il existe de nombreuses fonctionnalités offertes par le tableur, comme le tableau pivot par exemple. Cependant, l'utilisation de ces outils requiert là encore, de la part de l'utilisateur, une bonne compréhension du problème à résoudre mais également une bonne définition des dimensions du problème ainsi que les différentes variables exprimées selon ces dimensions.

1.1.1 Microsoft® Excel

Le programme Microsoft® Excel fait partie des références parmi les logiciels de type tableur. Il permet une gestion complète des tableurs et de leurs concepts mais il offre également d'autres fonctionnalités. En nous basant sur les explications définies dans (Gris, Février 2013) nous allons passer en revue quelques-unes de ces fonctionnalités importantes.

Pour commencer, attardons-nous sur ce qui est le plus important dans le cadre de ce document, la gestion des feuilles de calcul.

A son ouverture, le tableur Microsoft® Excel (dans sa version 2010), présente au centre de sa fenêtre principale, une page blanche constituée de lignes et de colonnes définissant une feuille de calcul. Celle-ci, ainsi que les autres éléments gérés par le tableur, sont regroupés dans un classeur qui peut d'ailleurs contenir plusieurs feuilles de calcul.

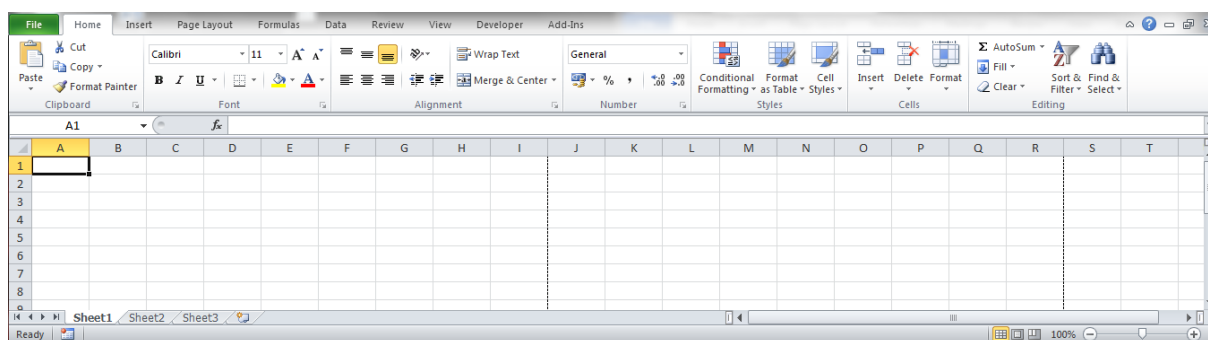


Figure 1 - Vue d'ensemble fenêtre principale Microsoft® Excel 2010

A partir de la feuille de calcul active, l'utilisateur a un accès direct aux cellules qui correspondent aux cases situées à l'intersection des lignes et des colonnes. C'est dans ces cellules que l'utilisateur a la possibilité d'encoder directement l'objet de son problème de calcul. En effet, les cellules peuvent contenir tant les données connues de l'utilisateur que les formules décrivant la manière de calculer les résultats recherchés. Sur base de ces données et de ces définitions de calcul, le tableur va calculer dynamiquement les valeurs des résultats et les afficher dans les cellules où sont définies les formules. Comme expliqué dans (Hainaut, 2005) ce calcul est réalisé de manière directionnelle, c'est-à-dire que le logiciel calcule les résultats sur base des données connues au préalable.

Pour distinguer les données brutes des formules, le logiciel requiert pour les secondes une syntaxe particulière commençant par le signe « égale » (=).

	A	B	C
1			
2		Tâches	Charge estimé (jour/personne)
3		Analyse architecture	50
4		Analyse Détaillé	75
5		Développement acces base de donnée	60
6		Développement gestion client	150
7		Développement gestion compte	140
8		Test gestion client	20
9		Test gestion compte	20
10		Grand total	=SUM(C3:C9)

	A	B	C
1			
2		Tâches	Charge estimé (jour/personne)
3		Analyse architecture	50
4		Analyse Détaillé	75
5		Développement acces base de donnée	60
6		Développement gestion client	150
7		Développement gestion compte	140
8		Test gestion client	20
9		Test gestion compte	20
10		Grand total	515

Figure 2 - Exemple encodage données brutes et formule résultat

Le tableur Excel permet la gestion de différents types de données : numériques, booléennes, chaîne de caractères, date et temps. Pour chacun de ces types, une série de fonctions pré-disponibles est utilisable dans la définition des formules. Par exemple, la fonction **SOMME**³ permet d'additionner l'ensemble des valeurs définies dans le groupe de cellules qui lui est référencé en paramètres. La fonction **GAUCHE**⁴ permet quant à elle d'extraire les n premiers caractères (en partant de la gauche) d'une chaîne de caractères fournie en paramètre de la fonction.

Comme nous pouvons le remarquer dans la formule de la Figure 1, les cellules sont référencées par une adresse. Cette adresse représente la position d'une cellule dans une feuille de calcul. Ainsi les cellules sont désignées par leur position colonne-ligne. La première cellule est donc désignée par l'adresse A1.

Lorsqu'elles sont utilisées dans une formule, les cellules peuvent être référencées selon trois manières différentes : par valeur relative, par valeur absolue, par nom. La Figure 3 nous en montre une illustration.

³ Dans la version anglaise du logiciel, il s'agit de la fonction **SUM**.

⁴ Dans la version anglaise du logiciel, il s'agit de la fonction **LEFT**.

val		fx		16	
	A	B	C	D	E
1	Donnée	Résultat			
2		16	=A2+4		
3			=A\$2+4		
4			=val+4		

val		fx		16	
	A	B	C	D	E
1	Donnée	Résultat			
2		16	20		
3			20		
4			20		

Figure 3 - Utilisation de référence de cellule

Le référencement d'une cellule par valeur relative est la méthode utilisée par défaut. Elle consiste à renseigner directement la position colonne-ligne de la cellule référencée. Le logiciel gère alors la position de cette cellule de manière relative à la cellule qui l'utilise. La position retenue n'est alors pas réellement la position colonne-ligne mais plutôt le chemin à parcourir pour atteindre la cellule référencée.

Dans la Figure 4, nous représentons ce principe. Si l'on considère les positions des cellules comme des couples (numéro de colonne, numéro de ligne) et les chemins comme des couples (nombre de colonnes à soustraire/à ajouter, nombre de lignes à soustraire/à ajouter), nous pouvons remarquer que lorsque la cellule B2 référence la cellule A1, la position relative retenue de la cellule A1 est en réalité le chemin (-1,-1). De même que, lorsque la cellule C2 référence la cellule D1 le chemin pour atteindre D1 à partir de C2 est le chemin (+1,-1).

	A	B	C	D
1	1 (1,1)	=A1 (2,1)	=D1 (3,1)	2 (4,1)
2	=A1 (1,2)	=A1 (2,2)	=D1 (3,2)	=D1 (4,2)
3	=A4 (1,3)	=A4 (2,3)	=D4 (3,3)	=D4 (4,3)
4	3 (1,4)	=A4 (2,4)	=D4 (3,4)	4 (4,4)

	A	B	C	D
1	1 (1,1)	1 (2,1)	2 (3,1)	2 (4,1)
2	1 (1,2)	1 (2,2)	2 (3,2)	2 (4,2)
3	3 (1,3)	3 (2,3)	4 (3,3)	4 (4,3)
4	3 (1,4)	3 (2,4)	4 (3,4)	4 (4,4)

Figure 4 - Schématisation de la gestion des références relatives

Le référencement d'une cellule par valeur absolue s'applique par l'utilisation du caractère spécial « \$ » devant la colonne et/ou devant la ligne de la référence colonne-ligne d'une cellule utilisée dans une formule. Le principe de ce type de référencement est qu'il désigne une cellule par son adresse réelle et plus en fonction de sa position par rapport à la cellule qui l'utilise.

Le référencement d'une cellule par un nom se base sur le même principe que le référencement par valeur absolue dans le sens où il identifie de manière unique une cellule. Son utilisation requiert que la cellule soit nommée au préalable à l'utilisation par son nom dans les formules qui y font référence. Comme le montre l'illustration Figure 5, ce nom est alors associé à l'adresse absolue de la cellule.

On notera au passage que, dans ce cas, l'adresse absolue est définie par le triplet « nom de feuille de calcul – colonne – ligne ». Cela s'explique par le fait que ce nom peut être utilisé dans des cellules situées dans d'autres feuilles de calcul.

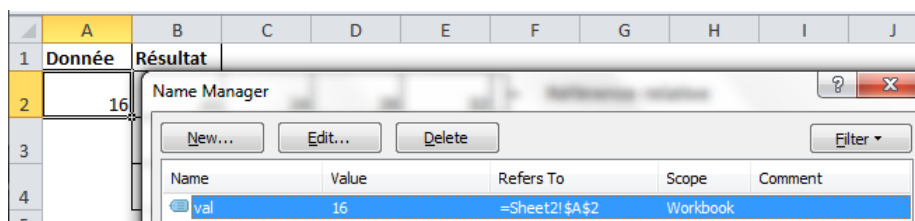


Figure 5 - Référencement d'une cellule par un nom

L'intérêt de chacune de ces possibilités de référencement réside dans la manière dont ces références sont gérées lors d'une action de copier-coller. En effet, dans pareilles situations, la référence sera ou non adaptée dynamiquement.

Dans le cas d'une référence par valeur relative, la valeur du chemin à parcourir reste fixe et la référence est à adapter dynamiquement. De sorte que, si la cellule B2 est copiée en C2 alors qu'elle référence la cellule A2, c'est-à-dire un chemin (-1,0), alors la cellule C2 contiendra, non pas la référence A2 mais bien B2 puis que le chemin (-1,0) reste inchangé.

Dans le cas d'une référence par valeur absolue ou par nommage par contre, la référence reste statique. Ainsi, lorsque la cellule B3 référence la cellule A2 et qu'elle est copiée en C3, la référence dans la nouvelle cellule reste A2.

Ces différentes situations sont représentées dans la figure suivante (Figure 6).

val		fx		16				
	A	B	C	D	E	F	G	H
1	Donnée	Résultat						
2	16	=A2+4	=B2+4	=C2+4	=D2+4			Référence relative
3		=\$A\$2+4	=\$A\$2+4	=\$A\$2+4	=\$A\$2+4			Référence absolue
4		=val+4	=val+4	=val+4	=val+4			Référence par nom

val		fx		16				
	A	B	C	D	E	F	G	H
1	Donnée	Résultat						
2	16	20	24	28	32			Référence relative
3		20	20	20	20			Référence absolue
4		20	20	20	20			Référence par nom

Figure 6 - Gestion des différents types de référence par l'action « copier/coller »

Comme nous pouvons le voir dans la formule définie dans la Figure 2, il est également possible de référer un groupe de cellules. Ce groupe est appelé une plage de cellules. Le référencement d'une plage est géré identiquement à celui que nous venons d'expliquer pour les cellules. La seule différence est que l'adresse d'une plage est constituée de deux adresses de cellule séparées par deux points (:).

Au niveau de la gestion des données introduites par les utilisateurs dans les cellules, le tableur Excel offre des possibilités de contrôle. Il est ainsi possible de définir parmi une liste de types de données standards (nombre entier, nombre décimal, liste, ...) un intervalle de valeur autorisé. Comme nous

pouvons le voir dans la Figure 7, lorsqu'un tel contrôle est mis en place, l'utilisateur est averti et stoppé par un message personnalisable lorsqu'il introduit une donnée non autorisée.

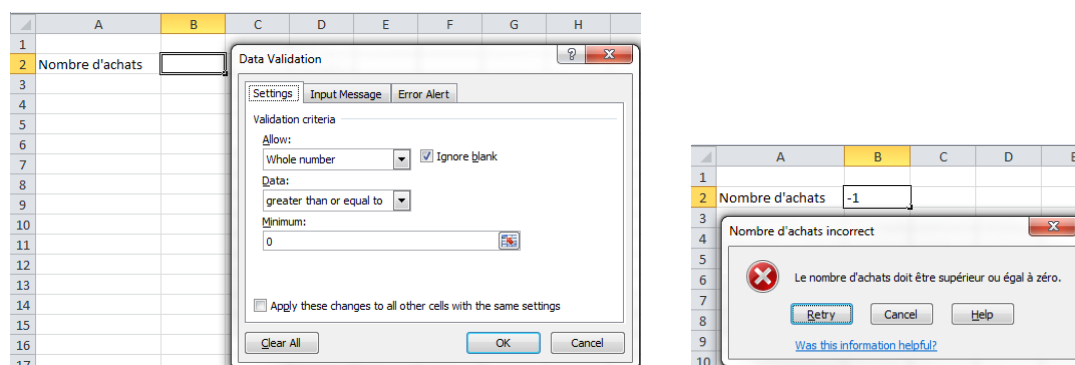


Figure 7 - Contrôle des données autorisées

Côté formules de calcul, comme déjà mentionné précédemment, Excel offre une panoplie de fonctions prédéfinies permettant diverses opérations. On retrouve notamment des fonctions réalisant des opérations mathématiques, de type booléen, statistique ou permettant la gestion de chaînes de caractères. Il y également des opérations plus particulières, mais fort utiles dans un contexte de gestion de données dimensionnées, comme par exemple les fonctions de recherche.

Les fonctions RECHERCHEV et RECHERCHEH⁵ permettent, un peu à la manière d'un annuaire, de retrouver la valeur « associée » à une valeur recherchée. Pour être opérationnelles, ces fonctions requièrent une plage de cellules contenant les données qui seront recherchées et retournées. Dans notre analogie il s'agit de l'annuaire. Nous retrouvons cette plage dans la Figure 8 en position A2:C6. Suivant la fonction utilisée, RECHERCHEV ou RECHERCHEH, le tableau sera disposé de manière verticale (V) ou horizontale (H) de telle sorte que l'index de recherche doit être défini en première colonne ou en première ligne. Enfin, le tableau doit être trié par ordre croissant. Pour utiliser ces fonctions, il suffit de passer en argument la valeur recherchée, la plage « annuaire » et le numéro de la colonne ou ligne correspondant à la position de la valeur associée dans le tableau. Un dernier paramètre permet d'activer ou désactiver une option de recherche qui permet de retourner tout de même une valeur associée lorsque la valeur recherchée n'est pas trouvée. Dans ce cas, la donnée d'index retenue sera la donnée immédiatement inférieure à la valeur recherchée.

	A	B	C	D	E
1	Article	Nom	Prix		
2	Art-1	Article-1	€ 5,00		
3	Art-2	Article-2	€ 3,00		
4	Art-3	Article-3	€ 12,00		
5	Art-4	Article-4	€ 10,00		
6	Art-5	Article-5	€ 6,00		
7					
8					
9	Article	Quantité	Prix/unité	Total/Article	
10	Art-1	6	=VLOOKUP(A10;A\$2:\$C\$6;3;FALSE)	€ 30,00	
11	Art-4	2	€	10,00	€ 20,00
12	Art-3	1	€	12,00	€ 12,00
13				€ 62,00	Total

	A	B	C	D	E
1	Article	Nom	Prix		
2	Art-1	Article-1	€ 5,00		
3	Art-2	Article-2	€ 3,00		
4	Art-3	Article-3	€ 12,00		
5	Art-4	Article-4	€ 10,00		
6	Art-5	Article-5	€ 6,00		
7					
8					
9	Article	Quantité	Prix/unité	Total/Article	
10	Art-6	6	=VLOOKUP(A10;A\$2:\$C\$6;3;TRUE)	€ 36,00	
11	Art-7	2	€	6,00	€ 12,00
12	Art-8	1	€	6,00	€ 6,00
13				€ 54,00	Total

Figure 8 - Utilisation de la fonction RECHERCHEV

⁵ Dans la version anglaise du logiciel, il s'agit des fonctions **VLOOKUP** et **HLOOKUP**

La fonction **SOMME**, dont nous avons déjà parlé est également utile dans un contexte de problème dimensionné. En effet, elle permet de réaliser des agrégations de données pour une série de cellules pouvant être regroupées en plage. Le tableur propose également d'autres fonctions d'agrégation et parmi celles-ci des agrégations conditionnelles. Leur fonctionnement repose sur le même principe que les agrégations plus classiques, mais au lieu d'appliquer l'opération d'agrégation sur l'ensemble des valeurs passées en paramètre, elles ne considèrent que les valeurs respectant la condition définie. Cela permet notamment de gérer des problèmes de calcul comme celui de l'exemple de la Figure 9 où le but recherché est de connaître la quantité d'articles vendus et le montant total des ventes par article sur base d'un ensemble de commandes. Cet exemple illustre l'utilisation qui peut être faite de la fonction **SOMME.SI**⁶

	A	B	C	D	E
8	Commande	000568			
9	Article	Quantité	Prix/unité	Total/Arti	
10	Art-1	6	€ 5,00	€ 30,00	
11	Art-2	7	€ 3,00	€ 21,00	
12	Art-3	1	€ 12,00	€ 12,00	
13				€ 63,00	Total
14	Commande	000569			
15	Article	Quantité	Prix/unité	Total/Arti	
16	Art-4	3	€ 10,00	€ 30,00	
17	Art-5	2	€ 6,00	€ 12,00	
18				€ 42,00	Total
19	Commande	000570			
20	Article	Quantité	Prix/unité	Total/Arti	
21	Art-1	6	€ 5,00	€ 30,00	
22	Art-3	2	€ 12,00	€ 24,00	
23	Art-4	2	€ 10,00	€ 20,00	
24				€ 74,00	Total
25					
26					
27	Article	Nombre vendu	Montant vendu par article		
28	Art-1	=SUMIF(\$A\$10:\$A\$23;A29;B\$10:\$B\$23)	=SUMIF(\$A\$10:\$A\$23;A29;D\$10:\$D\$23)		
29	Art-2				
30	Art-3				
31	Art-4				
32	Art-5				
33					

Figure 9 - Utilisation de la fonction **SOMME.SI**

D'un point de vue plus esthétique et visuel, le tableur offre également un large éventail de possibilités de mise en forme, tant au niveau du contenu des cellules que du rendu même de ces dernières. Il est ainsi possible, par exemple, de différencier les cellules « libellées » des cellules contenant les données de calcul ou bien de mettre en évidence les données importantes. Microsoft® Excel offre également la possibilité de définir des règles de formatage. Comme nous pouvons le voir dans la Figure 10, le principe est alors d'appliquer un format prédéfini ou personnalisé en fonction de la valeur contenue dans la ou les cellules concernées par ces règles. Dans ce cas, on parle de mise en forme conditionnelle. La mise en place de telles règles est possible uniquement pour les types de données prédéfinis tels que les valeurs numériques, les dates et les chaînes de caractères.

⁶ Dans la version anglaise du logiciel, il s'agit de la fonction **SUMIF**

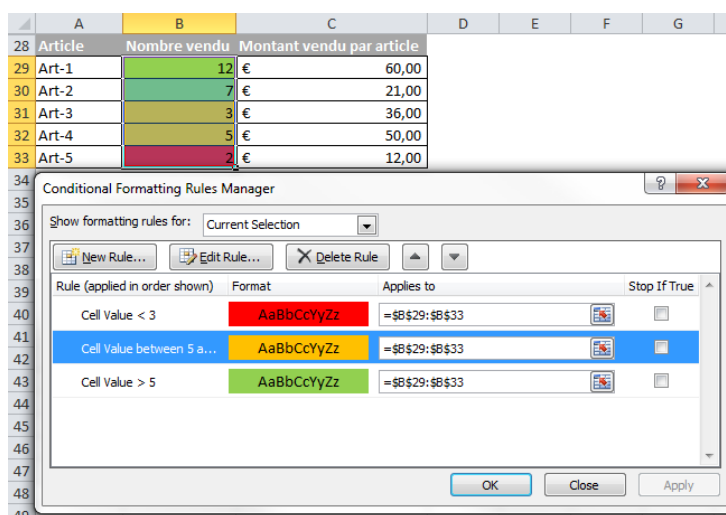


Figure 10 - Exemple d'utilisation de mise en forme conditionnelle

1.1.2 Autres logiciels grand public

Comme abordé dans l'introduction de cette section, il existe d'autres logiciels semblables au tableur Microsoft® Excel que ce soit au niveau des méthodes de travail ou du niveau d'accessibilité pour le grand public. Par exemple, LibreOffice Calc dont l'objectif est d'offrir les mêmes fonctionnalités qu'Excel mais dans une optique de logiciel libre et gratuit. Nous ne développerons pas plus ce point étant donné que LibreOffice Calc ne se différencie d'Excel, qui a été développé plus longuement ci-dessus, que par quelques subtilités.

1.2 Solutions logicielles avancées

Il existe ensuite des logiciels spécialisés, proposant des solutions avancées, habituellement utilisés dans le cadre de problèmes décisionnels.

Certains d'entre eux sont spécialisés dans la gestion de dimensions. Ils permettent de gérer de manière plus poussée les différentes dimensions. Par exemple, si on prend une base de données brute constituée de chiffres de vente, ce type de logiciel permet de réorganiser et de hiérarchiser les données sur base de la dimension temps (classement par trimestre, par mois...). Certains de ces logiciels se basent sur la technologie OLAP.

D'autres logiciels spécialisés abordent la traduction de modèle dans le tableur de base (style Excel). Il s'agit par exemple de ViTSL Editor et de son plugin Excel (ClassSheet - model-based, object-oriented design of spreadsheet applications, 2007).

Ces solutions logicielles avancées restent destinées à la gestion de très gros systèmes d'information ou ne sont utilisées que par des milieux professionnels très spécifiques. Elles sortent donc du domaine de cette étude qui tente de trouver des solutions pour le grand public ou de petites structures professionnelles.

1.3 Solutions méthodologiques : modèles de conception

Bien que la prise en main des outils tels que les tableurs soit relativement aisée pour des problèmes peu complexes, il n'est pas rare de découvrir des erreurs dans les différentes formules décrivant les feuilles de calcul. Si ces erreurs sont rares et faibles de conséquences pour des feuilles de calcul de petite taille, il n'en est rien pour des feuilles de grande taille. Dans ce cas, les conséquences peuvent mener à des erreurs de budget et à la prise de mauvaises décisions stratégiques pour les entreprises qui les utilisent par exemple. L'origine de ces erreurs peut être diverse : faute dans l'implémentation de formule, mauvaise manipulation de cellules,... Il y a cependant une origine plus problématique encore : la mauvaise compréhension du problème auquel est destinée la feuille de calcul. En effet, comme l'une des forces des tableurs est de permettre aux utilisateurs de rapidement décrire le problème en résultat, la mauvaise compréhension ou la description erronée d'un problème conduira à des calculs erronés, qui ne seront d'ailleurs pas toujours décelés. La compréhension mais aussi la formulation d'un problème posé sont donc primordiales dans la construction d'une feuille de calcul.

L'une des solutions envisagées pour faciliter et améliorer la compréhension des problèmes est de faire appel aux modèles de conception. Ces derniers ont en effet pour but premier de cerner et définir correctement tous les aspects d'un problème posé en faisant dans un premier temps abstraction des détails techniques de l'implantation dans le logiciel ou le tableur. Nous allons passer en revue quelques modèles de conception existants qui peuvent aider l'utilisateur dans cette première étape essentielle :

- Modèles « en étoile » et « en flocon » : ces modèles sont utilisés dans le contexte de modélisation de data warehouse dont l'objectif est de réaliser des agrégations de données, correspondant au concept de « fait » dans le modèle, en fonction de dimensions qui y sont reliées. La différence entre le modèle « flocon » et le modèle « étoile » réside dans la granularité des données liées aux dimensions. Par exemple, pour le modèle « en étoile », toutes les données d'une dimension sont sur un même niveau tandis qu'elles seront hiérarchisées dans le modèle « en flocon ».
- Modèle « Multidim » : il s'agit également d'un modèle de conception qui intervient dans le design de data warehouse mais dont l'objectif est de traiter encore plus explicitement la hiérarchie entre dimensions, et plus spécifiquement pour les dimensions spatiales et temporelles (Elzbieta, et al., 2008).
- Modèle de calcul de (Hainaut, 2005) : ce modèle est quant à lui plus tourné vers la gestion des problèmes de calcul dont la résolution peut être réalisée au moyen d'un tableur. Pour ce faire, le modèle se base sur le concept de « grandeur » qui se décline en « données » (informations fournies par les utilisateurs) ou en « résultats » (informations calculées par le tableur sur base des données, ils sont définis comme des relations entre grandeurs). Dans ce modèle la notion de dimension est également abordée et représentée par des grandeurs dimensionnées. C'est cette méthode que nous avons choisi d'utiliser comme point de départ pour notre étude, comme détaillé dans la section **Error! Reference source not found..**

2 Périmètre et objectifs du mémoire

Le présent document s'inscrit dans le cadre de la résolution de problèmes de calculs reposant sur la définition du problème au moyen d'un modèle de calcul et dont la solution est implantée dans un logiciel de type tableur.

Il a pour objectif de dresser et proposer un ensemble de règles de traduction facilement applicables afin de traduire les concepts décrits dans un modèle conceptuel - plus particulièrement celui introduit dans (Hainaut, 2005) - en élément physique d'un tableur et ce afin de faciliter l'implantation de modèle dans les tableurs.

Ces règles de traduction ont pour volonté d'être utilisables dans un processus de résolution de problèmes de calcul de taille moyennement complexe et solutionnées par un public dans un cadre domestique. Nous considérons dès lors des problèmes de calcul traduits dans un tableur de l'ordre d'environ 1000 cellules pour la définition des données et 1000 cellules pour la définition des résultats et résultats intermédiaires confondus. Nous n'envisagerons pas non plus des problèmes impliquant des données s'exprimant dans plus de 10 dimensions.

Ainsi les règles de traduction couvrent la traduction de modèles dimensionnés ne faisant pas appel à des sous-modèles. De plus, la modélisation ainsi que la traduction de contraintes inter-dimensions ainsi que la gestion de hiérarchie de dimensions ne seront pas fournies ici, cela devra faire l'objet d'une étude additionnelle du sujet.

Comme nous le présenterons ultérieurement, notre méthode de transformation repose sur trois niveaux d'abstraction (conceptuel, logique, physique) et est réalisée en deux phases. Il y a d'une part la traduction du niveau conceptuel au niveau logique et d'autre part la traduction du niveau logique au niveau physique.

Comme support à la définition de ces règles, nous déterminons également un ensemble de méta-modèles. Chacun d'entre eux a pour but d'offrir une vue d'ensemble des différents concepts rencontrés dans chacun des niveaux d'abstraction et impliqués dans nos règles de traduction.

3 Un problème de calcul

3.1 Étude de cas

Nous allons maintenant introduire un exemple de cas nécessitant l'utilisation d'une feuille de calcul. Cet exemple traite d'un suivi de projet dont le but est de piloter un projet de système d'information.

Cet exemple se base sur les exemples défini dans (Morley, 2012).

3.1.1 Gestion de projet :

Le chef de projet d'une PME souhaite réaliser le suivi d'un nouveau projet qui vient de lui être attribué. Pour ce faire, il souhaite disposer d'une feuille de calcul définissant le récapitulatif mensuel constitué des informations suivantes :

- Le temps passé sur une tâche,
- Le temps restant à effectuer pour terminer une tâche,
- L'avancement d'une tâche, défini comme la différence entre les deux dernières évaluations du reste à faire.

Il souhaite pouvoir visualiser ces valeurs pour chaque semaine du mois observé et voudrait également pouvoir disposer d'un total de ces valeurs pour le mois observé.

Une attention particulière est donnée aux valeurs d'avancement. En effet, lorsque pour une même tâche, la valeur d'avancement est inférieure au temps passé, cela peut-être le signe d'un problème dans le déroulement du projet. Pour ce genre de situation, le chef de projet aimerait pouvoir visualiser rapidement les tâches problématiques.

Afin de réaliser toutes ces observations, le chef de projet connaît la charge initialement estimée pour chaque tâche, ainsi que la répartition de ces tâches au sein de l'équipe de développement.

Il dispose également des comptes rendus hebdomadaires de chaque membre de l'équipe travaillant sur le projet à suivre. Ces comptes rendus reprennent par jour et par intervenant, le temps passé sur les tâches qui leurs sont assignées. On retrouve également dans ce compte rendu, l'estimation du reste faire de chacune de ces tâches au bout de la semaine écoulée.

Notons enfin que chacune des valeurs est exprimée en jour/personne.

Dans un souci de simplification du problème, nous faisons également l'hypothèse qu'une même tâche est réalisée par un même membre de l'équipe.

4 Point de départ : un modèle de calcul conceptuel

Comme nous avons pu le voir dans la section « méthodologie de conception » (1.3), il est important, voire primordial, d'aborder un problème de calcul en commençant par sa modélisation plutôt que par son implantation directe dans un tableur. Dans cette optique, un modèle de calcul peut être considéré comme le point de départ à la résolution de tel problème au moyen d'un tableur.

Partant de ce principe, il nous semble naturel de considérer également les modèles de calcul comme une base à la transformation d'un problème de calcul vers sa résolution dans un tableur.

Notre approche n'ayant pas pour prétention d'offrir une méthodologie de transformation universelle, nous nous limitons au choix d'un seul modèle de calcul parmi les modèles de conception brièvement mentionnés dans la section 1.3. Nous avons ainsi porté notre choix sur le modèle de calcul introduit dans (Hainaut, 2005).

Nous avons décidé de travailler sur base de ce modèle car il reste abordable pour les utilisateurs non informaticiens tout en étant proche du fonctionnement intrinsèque des outils que sont les tableurs. D'autre part, comme nous allons le voir ultérieurement, il traite de la problématique des données multidimensionnelles faisant partie de l'objet de notre travail.

Comme les concepts utilisés par ce modèle sont à la base de notre approche transformationnelle, il nous semble important de les aborder en détail. Dans la section suivante, nous allons ainsi définir les différents concepts de ce modèle de calcul tel qu'abordé dans (Hainaut, 2005).

Pour certains de ces concepts, nous prenons également la liberté d'augmenter, voire de redéfinir, leur signification et leur utilisation. De plus, nous introduisons certains nouveaux concepts dans ce modèle de calcul. Ces adaptations nous semblent pertinentes car certaines s'intègrent plus facilement dans notre processus de transformation et d'autres offrent une plus grande liberté dans la manière d'aborder la modélisation des dimensions. Pour chacun de ces points de modification nous précisons les changements, leur intégration dans le modèle conceptuel ainsi que leur raison d'être.

En support à cette description, nous introduisons également un méta-modèle du modèle de calcul. Ce dernier offre ainsi une vue d'ensemble sur les différents concepts et leurs relations. Comme nous l'aborderons dans la section 6, ce méta-modèle est également utilisé dans les règles de traduction constituant notre approche transformationnelle.

4.1 Les concepts

Dans (Hainaut, 2005), on décrit de manière générale un modèle comme étant constitué de grandeurs définissant soit des données, soit des résultats et des règles exprimant les résultats en fonction des données.

Le point de départ à tout modèle de calcul est le domaine d'application, soit le système à représenter. A partir de celui-ci, il y a lieu de repérer l'ensemble des concepts fondamentaux et importants à sa représentation. Ces concepts sont alors représentés dans le modèle abstrait par des **grandeurs** et décrits par un nom. Dans le contexte de modèle de calcul, ces concepts seront toujours des grandeurs mesurables dont la valeur associée sera de type numérique, logique ou qualitatif. On parle d'état déterminé du domaine lorsque chaque grandeur est définie par une valeur.

- Les grandeurs de type **numérique** sont utilisées pour exprimer des faits quantifiables. Par exemple, le nombre de jours/personne restant avant la fin d'un projet, le pourcentage d'avancement d'une tâche à réaliser pour un projet,...
- On utilisera des grandeurs de type **logique** pour exprimer des faits définis par deux états. La disponibilité ou l'indisponibilité d'une ressource pour réaliser une tâche d'un projet ; une tâche est-elle commencée ou pas encore.
- On parlera de grandeur de type **qualitatif** pour exprimer des faits dont les valeurs associées ont une signification particulière. Par exemple, dans un projet logiciel, il y a différents types de personnes ressources disponibles pouvant être affectées à un certain type de tâches. Un développeur code et réalise les tests unitaires d'une fonctionnalité d'un programme.

Afin de spécifier complètement les grandeurs d'un modèle, il est également nécessaire d'en définir le domaine de valeurs et son unité de mesure. Le domaine de valeurs est spécifié quant à lui par un type standard, un intervalle ou même un ensemble de valeurs explicitement énumérées. Afin de faciliter la lecture et la compréhension du modèle, on donne également une description succincte de ces grandeurs.

Parmi les concepts dégagés, certains sont dépendants les uns des autres. Ces relations peuvent être décrites dans le modèle par des **règles de définition**. Ces règles permettent d'obtenir la valeur d'une grandeur à partir des valeurs d'autres grandeurs. Elles sont construites avec une partie gauche, représentant la grandeur résultat, et une partie droite, constituée de l'expression définissant la grandeur résultat. Une fois implantées dans une feuille de calcul, ces règles de définition seront évaluées par le tableur de droite à gauche. Il est donc nécessaire de définir au préalable à la définition de ces règles quelles grandeurs sont connues a priori (il s'agit des données), et quelles sont les grandeurs dont on recherche la valeur (il s'agit des résultats).

En plus de définir un domaine de valeur, il est parfois nécessaire de définir des règles particulières appelées **contraintes**. Ces dernières sont décrites sous forme de règles et permettent de donner les états valides des grandeurs. Ces contraintes peuvent être utilisées tant pour contrôler et valider les données fournies pour les utilisateurs que pour contrôler la validité des résultats calculés pour le modèle.

Lors de la construction de modèles complexes, il est parfois intéressant d'introduire des grandeurs qui ne seront pas utiles pour l'utilisateur final du modèle mais qui aideront à sa compréhension, sa construction et sa maintenabilité. Ces grandeurs sont appelées **grandeurs internes** et sont différentes des concepts de grandeurs « données » et « résultats » déjà définis dans le sens où ces dernières participent uniquement au fonctionnement interne du modèle et ne doivent dès lors pas être connues des utilisateurs du modèle.

Dans certaines situations, il arrive que la définition d'une grandeur dépende d'un contexte particulier. On parle alors de **grandeurs à définition multiple**. Dans ce cas, pour définir la grandeur, on exprime les différents contextes sous forme de règles « Si » ainsi que la définition de la grandeur pour chacun des contextes. Ces définitions s'appellent des branches. Parmi celles-ci, il existe un type de branche particulier, appelé branche finale, qui permet d'exprimer la règle de définition d'une grandeur pour le contexte complémentaire à tous les autres. Ce contexte s'exprime alors avec la clause « sinon » au lieu de la clause « si » des autres contextes.

Bien que la spécification de **grandeurs à définition multiple** soit fort similaire à la définition de grandeurs plus « classiques », il est important de respecter trois règles essentielles pour garantir une définition correcte de ce type de grandeurs. Ces règles de « bonne pratique » sont les suivantes : la complétude, la non-ambiguïté et l'absence de branche morte.

Avec la complétude, on s'assure que la grandeur est définie dans tous les cas. Il est d'ailleurs recommandé de faire appel à la notion de branche finale qui en garantit cet aspect.

Par le respect de la non-ambiguïté, l'on veille à ce qu'il n'existe pas de situation telle que la grandeur soit définie plus d'une fois.

Une branche morte correspond à la définition d'une grandeur pour un contexte dont les conditions sont toujours fausses, quelles que soient les valeurs des données du modèle.

Bien que les tableurs soient utilisés le plus souvent pour la résolution de problèmes de calcul, ils offrent également la possibilité de résoudre des problèmes de type décisionnel. Pour ce faire, les tableurs utilisent des grandeurs non plus de type numérique mais de type logique. La modélisation

de tels systèmes et de leurs grandeurs dans un modèle de calcul consiste alors en la définition d'une base de règles qui s'appliquent ou non en fonction de la valeur des grandeurs qui les définissent.

Lors de l'exécution d'un modèle dans un tableur, il n'est pas rare que certaines données ne soient pas fournies ou même que l'évaluation d'une règle n'aboutisse pas et conduise à une erreur. Dans ce genre de situation, la majorité des tableurs disposent d'une valeur appelée « valeur d'exception ». Dans (Hainaut, 2005) on distingue deux de ces valeurs particulières : la valeur « absent » et la valeur « erreur ». Pour la première, il s'agit de représenter la situation où une grandeur du modèle n'a pu être calculée car une des données la définissant est manquante. Pour la deuxième, la valeur est employée pour une grandeur dont l'évaluation de la règle de définition tombe en erreur.

Ces différentes valeurs d'exception, lorsqu'elles surviennent, suivent les règles de propagation suivantes :

- Une grandeur B prendra la valeur « absent » lorsqu'une donnée A, utilisée dans la règle de définition de B, est manquante.
- Une grandeur C prendra la valeur « absent » lorsque la valeur « absent » a été assignée à une grandeur B qui est utilisée dans la règle de définition de C.
- Une grandeur C prendra la valeur « erreur » lorsque la valeur « erreur » a été assignée à une grandeur B qui est utilisée dans la règle de définition de C.
- En cas d'ambiguïté, la valeur « erreur » a priorité sur la valeur « absent »

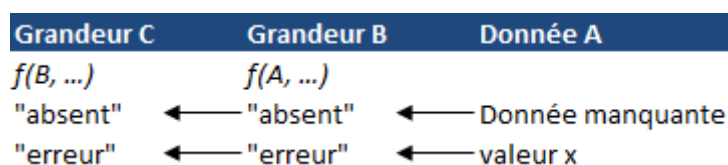


Figure 11 - Règles de propagations

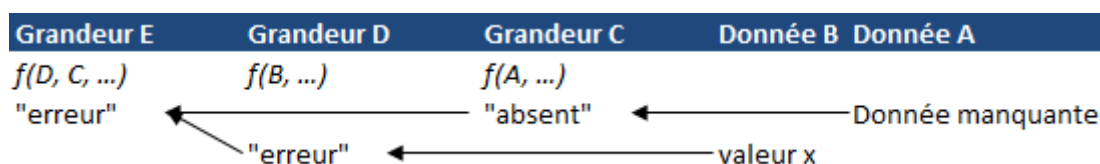


Figure 12 - Règles de propagations cas d'ambiguïté

Ces règles de propagation seront toujours d'application sauf dans les cas de grandeurs à définition multiple dont l'une des branches traite explicitement de la gestion de cette valeur d'exception et la traduit en une valeur normale. On dit alors que la valeur d'exception est absorbée par la règle de définition de la grandeur.

Les grandeurs définies par des règles logiques pourront également amener à l'absorption de valeurs d'exception car pour ces grandeurs ils existent des lois particulières liées à la logique non standard qui impliquent d'autres règles de propagation. Tout comme dans (Hainaut, 2005) nous n'aborderons pas ces règles particulières.

Dans certains cas, pour résoudre un problème de calcul, il est suffisant de se limiter à l'étude d'un état déterminé du domaine d'application. Par exemple, le suivi d'un projet pour une tâche de développement particulière à une date donnée. Dans cette situation, chacune des grandeurs du

modèle se voit attribuer une unique valeur représentative de l'état du domaine. Dans d'autres cas, par contre, il est nécessaire soit d'étudier un ensemble d'éléments du domaine partageant les mêmes caractéristiques (le suivi du projet à un moment donné pour toutes les tâches à réaliser), soit de s'intéresser à l'évolution des états du domaine (le suivi du projet par mois sur la durée du projet pour une tâche de développement particulière). Dans ces deux cas, il y a lieu d'aborder certaines caractéristiques du domaine d'application non plus pour un seul état déterminé mais plutôt pour une suite d'états définie selon un axe d'étude particulier du domaine (par exemple dans le cadre du suivi d'un projet informatique : la liste des tâches à réaliser, la liste des mois de la durée du projet). Dans ce contexte, l'utilisation d'une grandeur pour représenter un concept du domaine de valeur ne suffit plus puisqu'un même concept peut avoir plusieurs états. Les grandeurs du modèle se voient alors attribuer une suite de valeurs, on parle alors de **grandeur et de modèle dimensionné**.

Les modèles dimensionnés sont semblables aux modèles simples, ou non-dimensionnés, ils partagent dès lors les mêmes concepts. Cependant d'autres concepts supplémentaires sont à prendre en considération pour ces modèles.

L'un de ces concepts additionnels est la **dimension**, il permet de modéliser un axe d'étude particulier dans lequel est exprimé un ou des concepts du domaine d'application comme la liste des tâches d'un projet, la liste des mois suivis pour un projet, etc. Ce concept, proche de celui de grandeur, trouve sa particularité dans le fait qu'il est défini par un numéro index (ou indice) représentant de manière unique un élément particulier de l'axe d'étude qu'il représente. Une dimension sera définie dans le modèle par un nom et une description, cela de la manière semblable aux grandeurs. On définira également le nombre d'élément géré par la dimension. Par exemple pour une dimension de type date et plus précisément les jours de la semaine on peut définir le nombre d'élément à sept.

Associée au concept de dimension, la notion de grandeur dimensionnée permet de représenter plusieurs états déterminés d'un même élément du domaine d'application. Ces différents états sont ainsi envisagés selon un ou plusieurs axes d'étude qui sont représentés par les dimensions.

Le concept de grandeur dimensionnée se représente dans le modèle abstrait comme une grandeur classique mais indicée par la dimension dans laquelle elle est définie. Si l'on reprend notre exemple des jours de la semaine on définira la grandeur « nom des jours de la semaine » selon la dimension jour de la semaine.

Une telle grandeur représente en réalité non pas une seule grandeur mais une suite de grandeurs identiques dans leurs caractéristiques et dont la définition est faite pour chacune des valeurs d'index des dimensions dans lesquelles elle s'exprime.

Dans le contexte des grandeurs dimensionnées il est également possible de définir des règles de calcul basées sur des fonctions particulières, les fonctions agrégatives. Ces fonctions permettent de réaliser une opération mathématique d'agrégation tel quel la somme ou la moyenne sur l'ensemble des valeurs d'une grandeur dimensionnée. Dans le modèle, ces fonctions sont applicables sur des grandeurs dimensionnées et sont définis avec la ou les dimensions pour lesquelles les valeurs de la grandeur sont à considérer.

Dans certaines situations, comme le calcul d'un budget, il est nécessaire d'exprimer un résultat sur base de ses précédentes valeurs. Par exemple dans un budget, la valeur du montant disponible à un

mois donné est basée sur la valeur du montant disponible du mois précédent. Ce genre de situation peut être considéré dans le modèle conceptuel grâce au concept de règles de récurrence qui permet de définir une grandeur dimensionnée selon trois « sous-règles » de calcul. Il y a tout d'abord une règle d'initialisation dont l'objectif est de définir la valeur de la première instance d'une règle dimensionnée. Ensuite, nous retrouvons la règle de récurrence qui permet de définir la manière dont est calculée une grandeur dimensionnée sur base de sa valeur précédente. Enfin, il y a la règle de type final qui définit, dans une nouvelle grandeur la valeur terminale de la suite des calculs réalisés par la règle de récurrence.

4.2 Un outil d'analyse : le graphe des dépendances

Dans la construction du modèle de calcul, il peut être intéressant de visualiser de manière graphique les dépendances existantes entre les différentes grandeurs impliquées dans le modèle. En effet, lorsque l'on exprime dans le modèle une relation de type $A = f(B)$, on exprime que la grandeur A est définie en fonction de B. Cette grandeur A dépend donc directement d'une autre grandeur B. Par ailleurs, la suite de relation de type $A = f(B); B = g(C)$, exprime que la grandeur A dépend indirectement de la grandeur C.

Au moyen de la technique de visualisation graphique appelée graphe des dépendances, chacune des grandeurs est représentée par un sommet. Les relations de dépendance, quant à elles, sont représentées par un arc orienté dont le sommet d'arrivée est la grandeur définie et le sommet de départ, la grandeur participant à la détermination de celle-ci. Afin de garder une certaine lisibilité, toutes les relations ne sont pas représentées ainsi, seules les relations directes sont visibles dans ce graphe.

Les avantages d'utiliser cette visualisation sont multiples. En faisant ressortir la structure du modèle, cela permet de faire ressortir les grandeurs de type donnée, de type résultat, ainsi que les grandeurs internes.

Les données sont les grandeurs dont le sommet n'aboutit à aucun arc tandis que les résultats sont des grandeurs desquelles aucun arc ne prend son départ. Les grandeurs internes de même que les résultats sont identifiables par des sommets impliqués dans des arcs comme points de départ et d'arrivée.

Grace à cette visualisation il est également plus facile de choisir quelles grandeurs pourront être utilisées comme point de surveillance lors de l'utilisation de la feuille de calcul. Ces variables pourront alors être utilisées afin de prévenir un fonctionnement incohérent lors de l'instanciation du modèle.

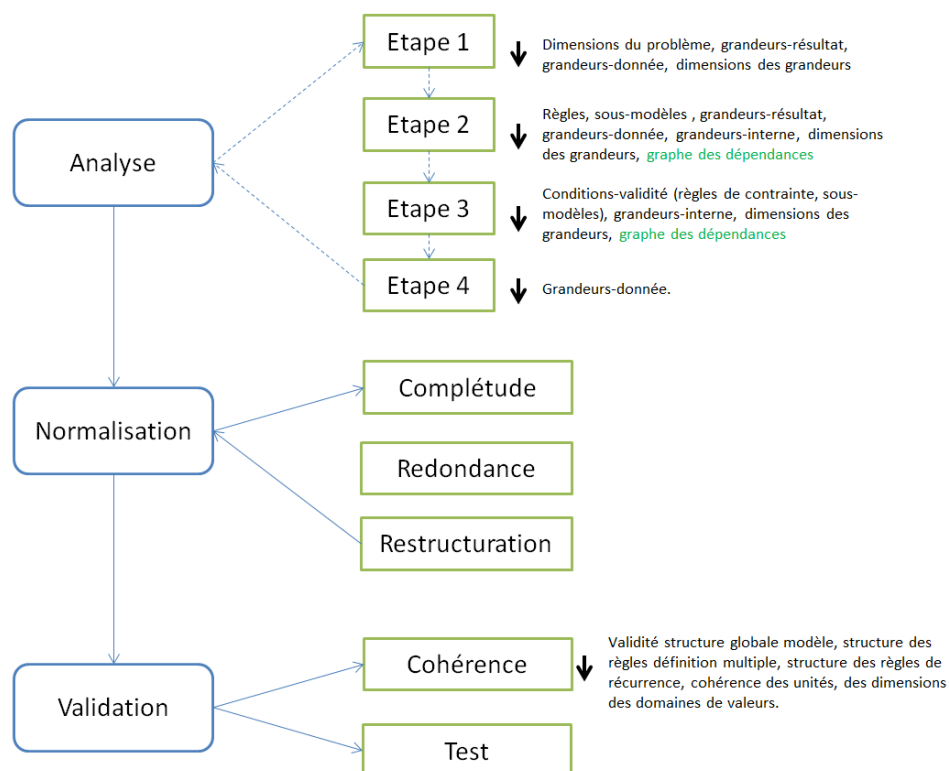
Un autre intérêt de cette visualisation graphique est qu'elle permet de repérer certaines erreurs de modélisation aboutissant à la présence de grandeurs qui ne sont ni utilisées ni définies. En effet, ces variables sont facilement identifiables car elles sont représentées par des sommets ne faisant partie d'aucun arc.

Dans un contexte de grandeurs dimensionnées, ce graphe permet également la détection de relations récursives. Celles-ci sont repérables car elles définissent un circuit⁷, ce qui traduit que chacune de ces grandeurs dépend d'elle-même que ce soit directement ou indirectement.

Pour toutes ces raisons, ce graphique des dépendances est un outil précieux lors de la construction du modèle de calcul.

4.3 La méthodologie

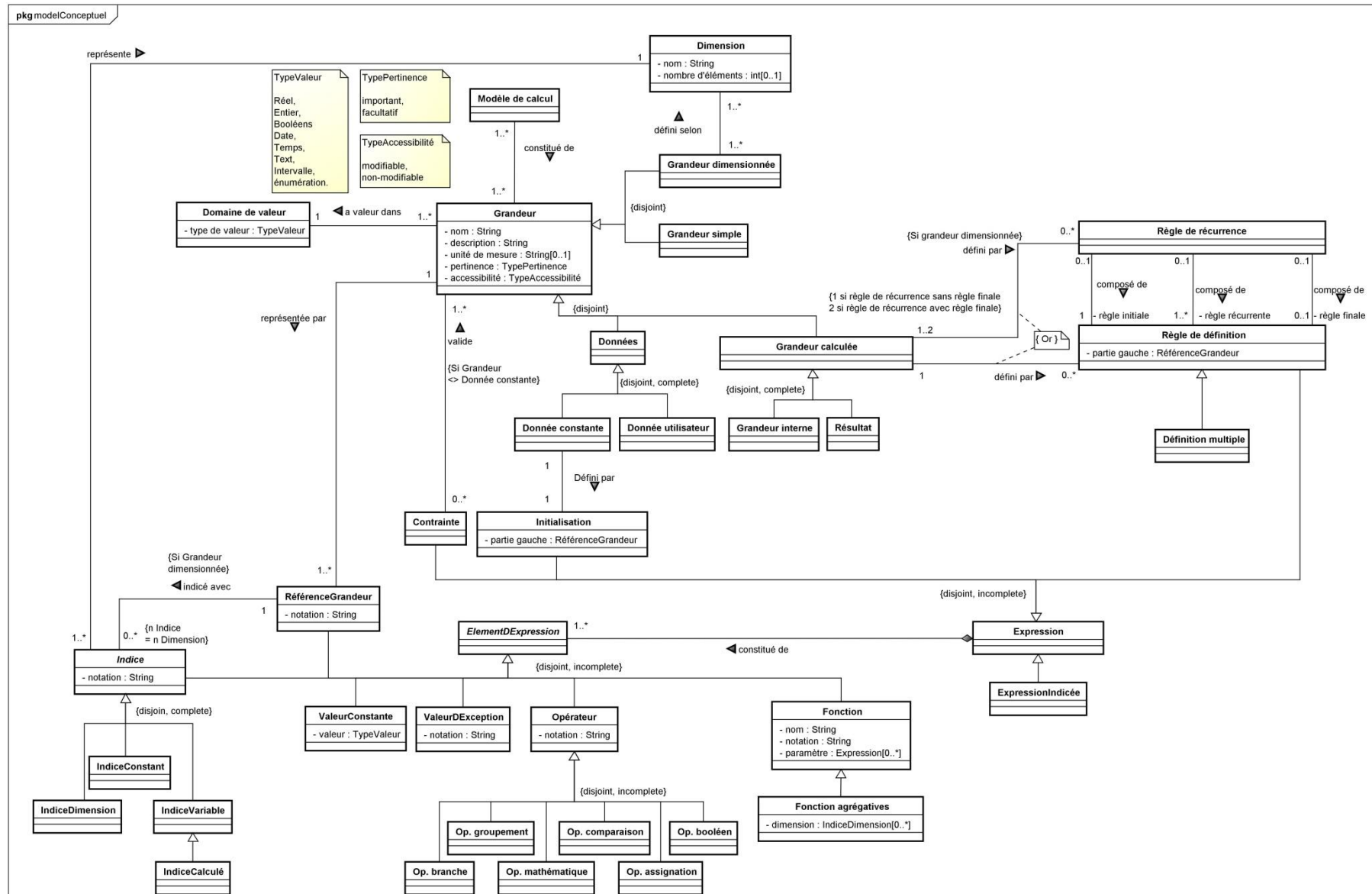
Outre la définition des différents concepts du modèle de calcul proposé dans (Hainaut, 2005) et que nous venons de passer en revue, (Hainaut, 2005) propose une méthodologie de conception pour traduire un problème, ainsi que les caractéristiques du domaine d'application y attachant, en un modèle de calcul correct, robuste, évolutif et surtout implantable dans des feuilles de calcul d'un tableur quelconque. Comme ce modèle de calcul est au cœur des solutions implantées dans les tableurs et qu'il forme le point de départ aux règles de traduction faisant l'objet de notre étude, il nous semble important de reprendre ici dans un schéma récapitulatif les différentes étapes envisagées.



4.4 Vue d'ensemble

Le schéma suivant représente une vue globale de l'ensemble de éléments impliqués dans la construction du modèle conceptuel.

⁷ Def. (Leclercq, 2008) : Un circuit d'un graphe orienté étant défini comme une suite d'arcs successifs dont le sommet d'origine du premier arc coïncide avec l'extrémité du dernier.



4.5 Application concrète

Nous définissons ici un modèle conceptuel selon les critères et méthodes définies dans (Hainaut, 2005). Ce modèle de calcul est un exemple de solution proposé par Mr Michaël Petit pour une gestion de suivit de projet semblable au cas d'exemple présenté précédement.

Dimensions

$P: \{P_1, \dots, P_{nb\text{-}periodes}\}$: Les périodes

$T: \{T_1, \dots, T_{nb\text{-}tâches}\}$: Les tâches

$R: \{R_1, \dots, R_{nb\text{-}ressources}\}$: Les ressources

Données

TempsPresté_{R,T,P}: Réel (heure) – Le temps presté par la ressource R, sur la tâche T, pendant la périod P

ResteAFaire_{R,T,P}: Réel (heure) – Le reste à faire pour la ressource R, sur la tâche T, à la fin de la périod P

ChargeAffectée_{R,T,P}: Réel (heure) – La charge affectée à la ressource R, sur la tâche T, pour la périod P

ChargeInitiale_T: Réel (heure) – La charge initiale de la tâche T

Contraintes

TempsPresté_{R,T,P} ≥ 0

ResteAFaire_{R,T,P} ≥ 0

ChargeAffectée_{R,T,P} ≥ 0

ChargeInitiale_T = ChargeAffectée_T

Résultats de période

Avancement_{R,T,P}: Réel (heure) – L'avancement de la ressource R, sur la tâche T, pendant la période P

Performance_{R,T,P}: Réel (%) – La performance de la ressource R, sur la tâche T, pendant la période P

EvolutionCharge_{R,T,P}: Réel (heure) – L'évolution de la charge restante de la ressource R, sur la tâche T, pendant la période P

%EvolutionCharge_{R,T,P}: Réel (%) – L'évolution en pourcents de la charge restante de la ressource R, sur la tâche T, pendant la période P

Vitesse_{R,T,P}: Réel (%) – La vitesse de la ressource R, sur la tâche T, pendant la période P (= l'évolution en pourcents de la charge restante)

Résultats cumulés par période

AvancementCumulé_{R,T,P}: Réel (heure) – L'avancement de la ressource R, sur la tâche T, à la période P, depuis le début du projet

%AvancementCumulé_{R,T,P}: Réel (%) – Le pourcentage d'avancement cumulé de la ressource R, sur la tâche T, à la période P, depuis le début du projet

TempsPrestéCumulé_{R,T,P}: Réel (heure) – Le temps presté depuis le début du projet par la ressource R, sur la tâche T, calculé à la période P

%UtilisationCumulé_{R,T,P}: Réel (%) – Le pourcentage d'utilisation de la ressource R, sur la tâche T, à la période P, depuis le début du projet

EvolutionChargeCumulée_{R,T,P}: Réel (heure) – L'évolution de la charge restante de la ressource R, sur la tâche T, à la période P depuis le début du projet

%EvolutionChargeCumulée_{R,T,P}: Réel (%) – Le pourcentage d'évolution de la charge restante de la ressource R, sur la tâche T, à la période P, depuis le début du projet

PerformanceCumulée_{R,T,P}: Réel (%) – La performance de la ressource R, sur la tâche T, à la période P, depuis le début du projet

VitesseCumulée_{R,T,P}: Réel (%) – La vitesse de la ressource R, sur la tâche T, à la période P, depuis le début du projet (= l'évolution en pourcents de la charge restante)

Résultats agrégés

ChargeAffectée_{R,T}: Réel (heure) – La charge totale affectée à la ressource R, sur la tâche T

ChargeAffectée_T: Réel (heure) – La charge totale affectée à la ressource R, sur la tâche T

Toutes les agrégations par tâches et ressources sont possibles. Par exemple: Avancement_{R,P}, Avancement_{T,P}, Avancement_P, ...

Règles

$$\text{ChargeAffectée}_{R,P} = \sum_T \text{ChargeAffectée}_{R,T,P}$$

$$\text{ChargeAffectée}_{T,P} = \sum_R \text{ChargeAffectée}_{R,T,P}$$

$$\text{ChargeAffectée}_T = \sum_P \text{ChargeAffectée}_{T,P}$$

$$\text{ChargeAffectée}_P = \sum_T \text{ChargeAffectée}_{T,P}$$

$$\text{ResteAFaire}_{T,P} = \sum_R \text{ResteAFaire}_{R,T,P}$$

$$\text{ResteAFaire}_{R,P} = \sum_T \text{ResteAFaire}_{R,T,P}$$

$$\text{ResteAFaire}_P = \sum_R \text{ResteAFaire}_{R,P}$$

$$\text{Avancement}_{R,T,P} = \text{ResteAFaire}_{R,T,P-1} - \text{ResteAFaire}_{R,T,P} \quad (P = 2 \dots \text{nb-periodes})$$

$$\text{Avancement}_{R,T,P} = \text{ChargeAffectée}_{R,T} - \text{ResteAFaire}_{R,T,P} \quad (P = 1)$$

$$\text{AvancementCumulé}_{R,T,P} = \text{Avancement}_{R,T,P} \quad (P=1)$$

$$\text{AvancementCumulé}_{R,T,P} = \text{AvancementCumulé}_{R,T,P-1} + \text{Avancement}_{R,T,P} \quad (P=2 \dots \text{nb-periodes})$$

$$\text{AvancementCumulé}_{R,P} = \sum_T \text{AvancementCumulé}_{R,T,P}$$

$$\text{AvancementCumulé}_{T,P} = \sum_R \text{AvancementCumulé}_{R,T,P}$$

$$\text{AvancementCumulé}_P = \sum_T \text{AvancementCumulé}_{T,P}$$

$$\% \text{Avancement}_{R,T,P} = \text{Avancement}_{R,T,P} / \text{ResteAFaire}_{R,T,P-1} \quad (P = 2 \dots \text{nb-periodes})$$

$$\% \text{Avancement}_{R,T,P} = \text{Avancement}_{R,T,P} / \text{ChargeAffectée}_{R,T} \quad (P = 1)$$

$$\% \text{Avancement}_{T,P} = \text{AvancementCumulé}_{T,P} / \text{ChargeAffectée}_{T,P}$$

$$\% \text{Avancement}_{R,P} = \text{AvancementCumulé}_{R,P} / \text{ChargeAffectée}_{R,P}$$

$$\% \text{Avancement}_P = \text{AvancementCumulé}_P / \text{ChargeAffectée}_P$$

$$\text{TempsPrestéCumulé}_{R,T,P} = \text{TempsPrestéCumulé}_{R,T,P-1} + \text{TempsPresté}_{R,T,P} \quad (P = 2 \dots \text{nb-periodes})$$

$$\text{TempsPrestéCumulé}_{R,T,P} = \text{TempsPresté}_{R,T,P} \quad (P = 1)$$

$$\text{TempsPrestéCumulé}_{T,P} = \sum_R \text{TempsPrestéCumulé}_{R,T,P}$$

$$\text{TempsPrestéCumulé}_{R,P} = \sum_T \text{TempsPrestéCumulé}_{R,T,P}$$

$$\% \text{Utilisation}_{R,T,P} = \text{TempsPresté}_{R,T,P} / \text{ChargeAffectée}_{R,T,P} \text{ SI } \text{ChargeAffectée}_{R,T,P} \neq 0$$

UNDEF SINON

%Utilisation_{R,P} =

Vitesse_{R,T,P} = Avancement_{R,T,P} / TempsPresté_{R,T,P} SI TempsPresté_{R,T,P} ≠ 0

UNDEF SINON

Performance_{R,T,P} = ResteAFaire_{R,T,P-1} / (TempsPresté_{R,T,P} + ResteAFaire_{R,T,P})

SI TempsPresté_{R,T,P} + ResteAFaire_{R,T,P} ≠ 0

UNDEF SINON (P = 2 ... nb-periodes)

Performance_{R,T,P} = ChargeAffectée_{R,T} / (TempsPresté_{R,T,P} + ResteAFaire_{R,T,P})

SI TempsPresté_{R,T,P} + ResteAFaire_{R,T,P} ≠ 0

UNDEF SINON (P = 1)

EvolutionCharge_{R,T,P} = TempsPresté_{R,T,P} – Avancement_{R,T,P}

%EvolutionCharge_{R,T,P} = (TempsPresté_{R,T,P} – Avancement_{R,T,P}) / ChargeAffectée_{R,T}

%AvancementCumulé_{R,T,P} = AvancementCumulé_{R,T,P} / ChargeAffectée_{R,T}

EvolutionChargeCumulée_{R,T,P} = TempsPrestéCumulé_{R,T,P} – AvancementCumulé_{R,T,P}

%EvolutionChargeCumulée_{R,T,P} = EvolutionChargeCumulée_{R,T,P} / ChargeAffectée_{R,T}

%UtilisationCumulé_{R,T,P} = TempsPrestéCumulé_{R,T,P} / ChargeAffectée_{R,T}

PerformanceCumulée_{R,T,P} = ChargeAffectée_{R,T} / (TempsPrestéCumulé_{R,T,P} + ResteAFaire_{R,T,P})

VitesseCumulée_{R,T,P} = AvancementCumulé_{R,T,P} / TempsPrestéCumulé_{R,T,P}

5 Concrétisation dans un tableur : un modèle physique

Le modèle physique que nous présentons ici est défini sur base des différents concepts génériques géré par les tableurs et plus spécifiquement pour le tableur Microsoft® Excel. Il a pour vocation d'identifier l'ensemble des concepts à prendre en considération lors de l'implantation d'une solution de problème de calcul dans un tableur. Dans cette définition, nous considérons la situation où les concepts gérés par le tableur sont dans un état stable. C'est-à-dire un état dans lequel le tableur a réalisé l'ensemble des calculs demandés.

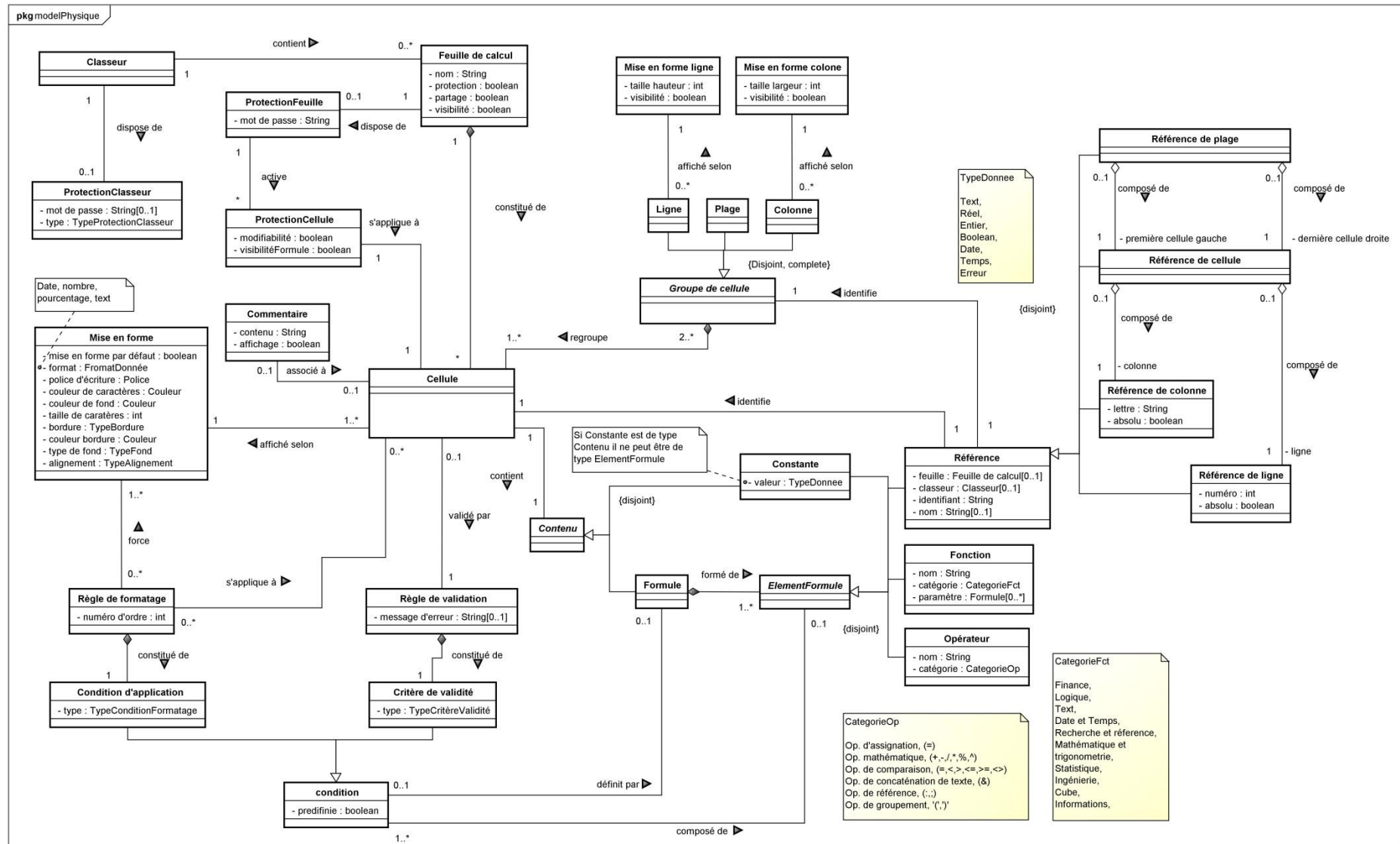
5.1 Les concepts

Bon nombre des concepts génériques gérés par les tableurs ont déjà été partiellement introduits dans la section **Error! Reference source not found..**

Sans grande surprise nous retrouvons l'élément central qu'est la cellule. Partant de ce concepts nous pouvons remarquer dans la section suivante l'ensemble de éléments permettant de la gérer.

5.2 Vue d'ensemble

Le schéma suivant représente un vue global de l'ensemble de éléments impliqués dans la construction du modèle physique.



6 Du modèle de calcul conceptuel aux feuilles de calculs

Jusqu'à présent, nous avons abordé la manière dont un problème de calcul peut-être décrit d'un point de vue conceptuel. Nous avons également détaillé de quelle façon ce genre de problème peut être défini dans un programme de type tableur afin de calculer les résultats recherchés. Bien qu'il soit possible de traduire directement un modèle conceptuel en une implantation directe dans le tableur de son choix, ce travail reste souvent intuitif et basé sur des règles de bonne pratique.

Dans cette section, nous présentons une méthode dont l'objectif est d'être plus systématique. Cette méthode s'inspire de ce qui se fait habituellement en ingénierie logiciel et repose sur trois niveaux d'abstractions pour l'expression d'un problème de calcul.

Comme nous pouvons le voir dans la Figure 13, le modèle de calcul précédemment abordé représente le niveau le plus abstrait, le niveau conceptuel. Au plus proche du tableur cible, on retrouve le modèle physique, présenté dans la section précédente. Il définit le niveau le plus concret juste avant le tableur. Le niveau logique est un niveau intermédiaire entre le modèle conceptuel et le modèle physique. Pour ce dernier, nous introduisons un nouveau modèle dont l'objectif est de pouvoir traiter de manière abstraite la disposition des éléments du modèle conceptuel tel qu'on pourrait l'envisager dans la définition de l'interface utilisateur d'un logiciel (Ramdoyal, 2012-2013).

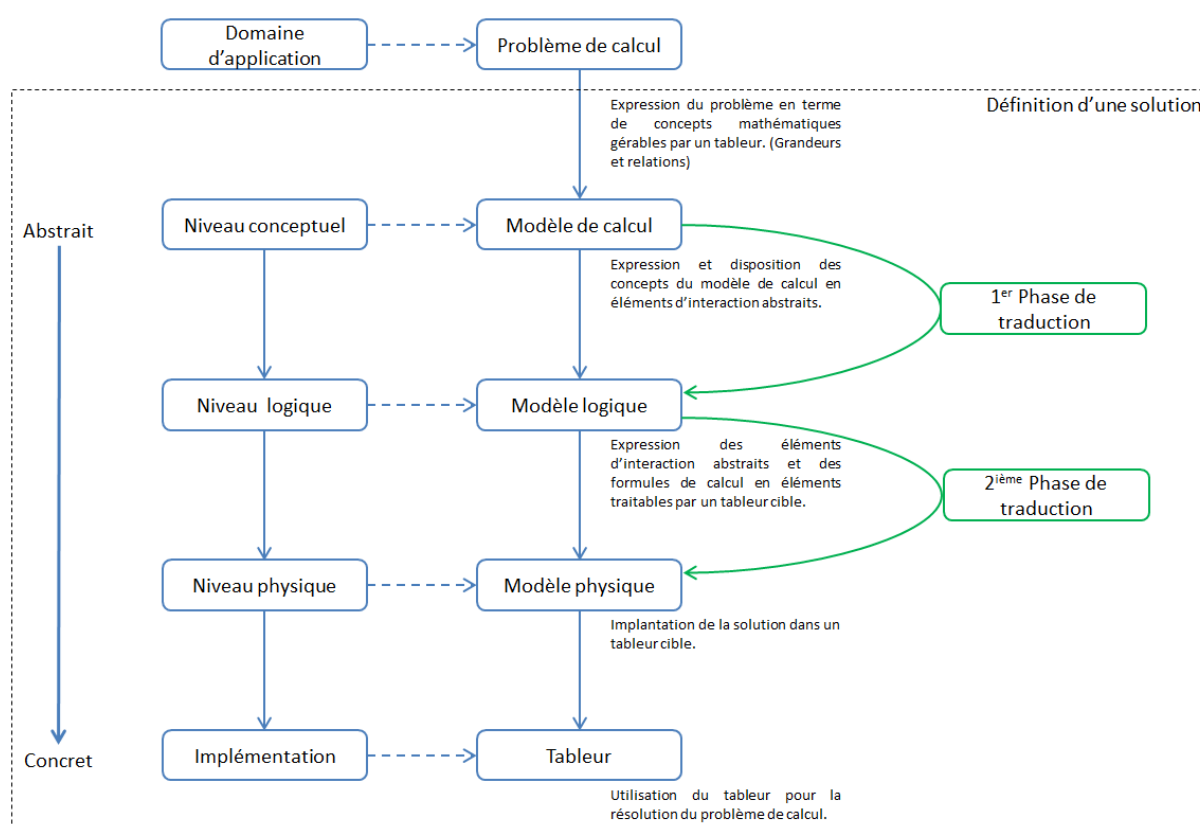


Figure 13 - Niveaux d'abstractions, résultats associés et phases de traduction

Partant de ces différents niveaux et modèles associés, nous proposons une transformation d'un problème de calcul vers sa résolution dans un tableur en deux phases de traduction. L'une permet la transition entre modèle de calcul conceptuel et modèle logique tandis que l'autre permet la traduction du modèle logique en modèle physique facilement implantable dans un tableur cible.

Dans les sections suivantes, nous allons détailler les concepts constituant notre modèle logique pour ensuite présenter les différentes règles de traduction impliquées dans nos deux phases de traduction.

6.1 Élément intermédiaire : un modèle logique

Le principal objectif de notre modèle logique est donc de permettre aux développeurs de solution calculée dans un tableur, de se focaliser sur l'aspect visuel souhaité. Si nous faisons le parallèle avec la construction d'un logiciel, cette étape permet de définir l'interface utilisateur. Nous partons pour cela de ce que l'on appelle dans (Hainaut, 2005) et (Petit, 2009-2010) une **maquette**. Dans ces ouvrages, il est question de maquette concrète (feuille de papier) sur laquelle on dispose directement les grandeurs du modèle de calcul. Dans notre modèle logique, il s'agit plutôt d'une entité abstraite qui regroupe l'ensemble de tous les éléments qui serviront à l'interaction utilisateur lors de l'utilisation de la solution de calcul implantée dans le tableur.

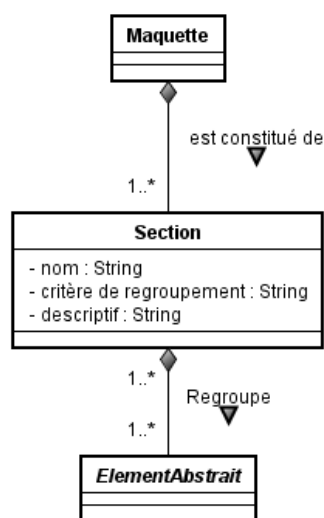


Figure 14 - Maquette et Section

Une **maquette** contient ainsi un ensemble d'éléments abstraits regroupés en **section**. Bien que l'élément de type **section** ne soit pas impliqué directement dans l'interaction avec les utilisateurs, il a pour vocation de structurer le groupement des différents éléments abstraits. On peut par exemple opter pour un regroupement par raison d'être : les données, les résultats internes, les résultats. Ou bien encore par but utilisateur : vue client, vue stock et article, vue vente. Par ce fait, un même élément abstrait pourra être défini dans différentes **sections**. Ce sera le cas par exemple pour notre regroupement par but utilisateur où l'on fera appel au *nombre total d'articles vendus* dans les sections « vue stock et article » et « vue vente ». On évitera cependant de dupliquer les éléments abstraits relatifs à la saisie de données pour assurer l'intégrité des données et faciliter l'utilisation de futures feuilles de calcul. Afin de documenter les choix posés lors de la construction du modèle physique, on veillera à préciser le critère de regroupement envisagé pour une section. On détaillera

également la raison d'être de la section. La Figure 15 nous montre ce que pourrait être la définition des sections selon les différents buts utilisateur.

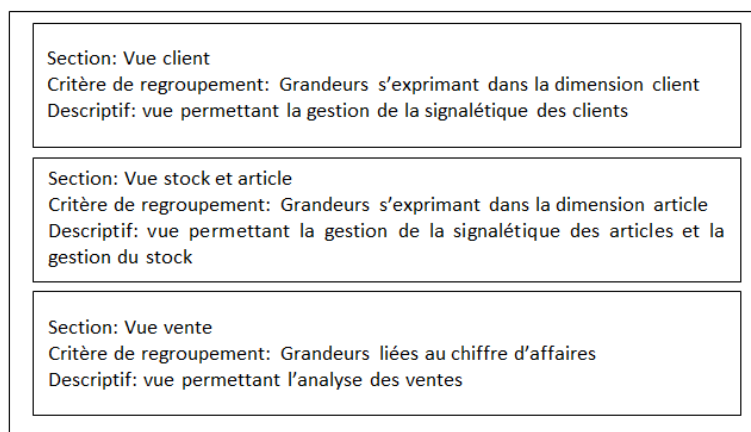


Figure 15 - Définitions des sections dans une maquette

Une fois les différentes sections déterminées, il reste à organiser les éléments de la section tels qu'ils seront définis dans le tableur physique. Pour cela, nous introduisons les **ElementDeGroupe**. Il s'agit d'éléments abstraits affichables pouvant servir à définir des sous-sections dans les sections, mais également à définir plus finement le positionnement des différents éléments qui seront affichés. Il en existe trois types : **GroupeDeProfondeur**, **GroupeHorizontal**, **GroupeVertical**. Chacun de ces groupes a pour vocation de gérer la disposition des éléments qu'il contient selon l'axe référencé dans son nom. Nous en donnons une illustration dans la Figure 16. Comme nous pouvons le voir, chaque groupe est défini par un nom identifiant et chaque élément qu'il contient est disposé selon un numéro d'ordre et suit l'axe de positionnement dans lequel il s'exprime.

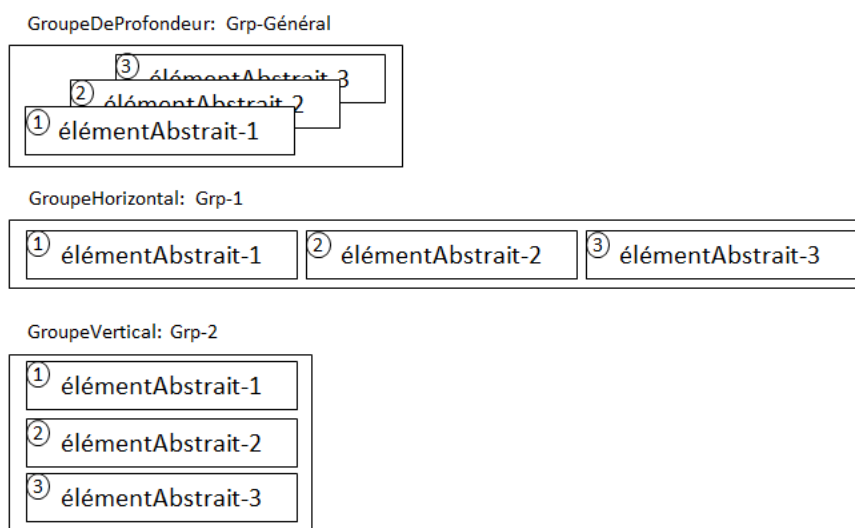


Figure 16 - élémentDeGroupe : **GroupeDeProfondeur**, **GroupeHorizontal**, **GroupeVertical**

La définition de ces trois groupes correspond ainsi aux trois dimensions gérées habituellement par les tableurs à savoir respectivement : les feuilles de calcul, les colonnes et les lignes. Il en découle ainsi la contrainte que les éléments de type **GroupeHorizontal**, **GroupeVertical** ne peuvent contenir de **GroupeDePofondeur**. Par ailleurs, une section contiendra toujours un **GroupeDePofondeur** regroupant l'ensemble des éléments affichage de la dite section.

Ces éléments ont également la possibilité d'être définis par un critère de répétition. Lorsque ce critère est déterminé, cela implique que le groupe représente en réalité plusieurs copies du groupe. Le nombre de copies étant déterminé par la définition du critère de répétition. Cette propriété permet notamment de gérer les grandeurs dimensionnées. En effet, le critère de répétition peut s'exprimer selon une dimension ou dans l'absolu par un nombre de répétitions. Dans ce cas, à chaque instance d'un groupe répété correspondra une instance de la dimension. Par ce moyen, les éléments du groupe défini selon cette dimension seront affichés dynamiquement en fonction des valeurs de cette dimension. Les éléments non liés à cette dernière seront, quant à eux, répétés tels quels sans modification. Tous ces éléments seront dupliqués selon le sens d'agencement défini dans le groupe qui les contient.

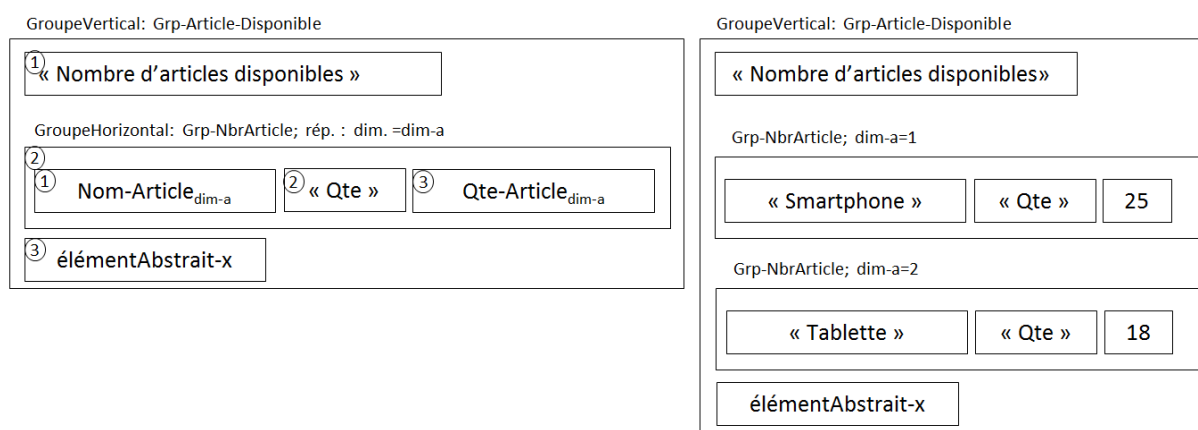


Figure 17 - ElementDeGroupement illustration de la propriété de répétition

Dans la définition de l'agencement des différents éléments abstraits, il est parfois nécessaire de définir une séparation bien distincte entre les éléments. C'est le cas par exemple lorsque l'on souhaite définir différentes sous-parties dans une même feuille de calcul. L'élément abstrait de type **Séparateur** permet cette gestion en consommant un numéro d'ordre au sein d'un **ElementDeGroupement**. Dans la Figure 18 nous représentons l'utilisation d'un **Séparateur** ainsi que sa matérialisation dans une feuille de calcul.

GroupeVertical: Grp-VenteArticle									
①	élémentAbstrait-A								
②	séparateur								
③	élémentAbstrait-B								
④	séparateur								
⑤	élémentAbstrait-C								

	A	B	C	D	E	F	G	
1	Prix Article							
2	Ecran plat			€ 65,00				
3	Ordinateur			€ 20,00				
4	Smartphone			€ 35,00				
5	Tablette			€ 28,00				
6								
7								
8								
9	Nom-1	Prénom-1	Qte Achat	2	1	0	0	
10			Mnt Achat	€ 130,00	€ 20,00	€ -	€ -	€
11	Nom-2	Prénom-2	Qte Achat	0	0	0	0	€
12			Mnt Achat	€ -	€ -	€ -	€ -	€
13	Nom-3	Prénom-3	Qte Achat	0	0	0	0	€
14			Mnt Achat	€ -	€ -	€ -	€ -	€
15	Nom-4	Prénom-4	Qte Achat	50	0	0	0	€
16			Mnt Achat	€ 3.250,00	€ -	€ -	€ -	€
17	Nom-4	Prénom-4	Qte Achat	0	0	0	0	€
18			Mnt Achat	€ -	€ -	€ -	€ -	€
19								
20	Total			Qte Achat	52	1	0	0
21				Mnt Achat	€ 3.380,00	€ 20,00	€ -	€ -

Figure 18 - Utilisation de séparateur

Dans d'autres situations, on peut souhaiter un alignement entre des éléments abstraits définis dans des éléments de groupement différents. Comme nous pouvons le voir pour les données « Qte

Achat » et « MntAchat » de la Figure 18 ce sera le cas lorsque l'on souhaite visualiser des données selon un agencement tabulaire avec des libellés de ligne et de colonne. Pour gérer ce genre de disposition il est possible d'utiliser un **lien d'alignement**. Ce concept permet en effet de fixer la position d'un élément abstrait par rapport à un autre élément abstrait. Nous en donnons une illustration dans la Figure 19. Concrètement on définira un d'alignement en précisant d'une part l'élément abstrait dont la position est la référence et d'autre part l'élément abstrait pour lequel on souhaite fixer l'alignement sur la référence. On précisera également quel type d'alignement est concerné : vertical, horizontal, de profondeur.

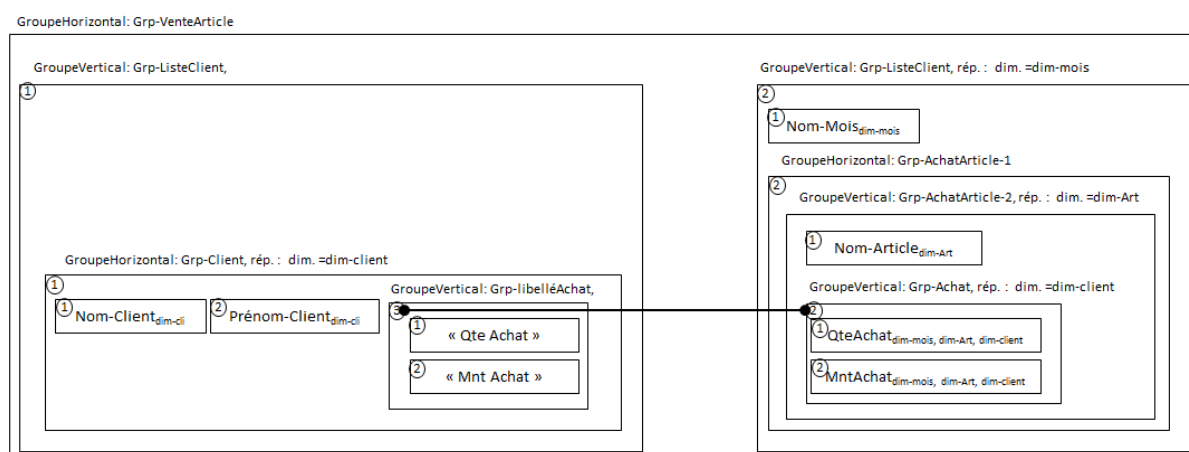


Figure 19 - Illustration lien d'alignement

Notons enfin que l'utilisation de liens d'alignement se limite aux *ElementAbstrait* définis comme affichables et qu'il faut que ces éléments soient définis dans des éléments de groupement différents.

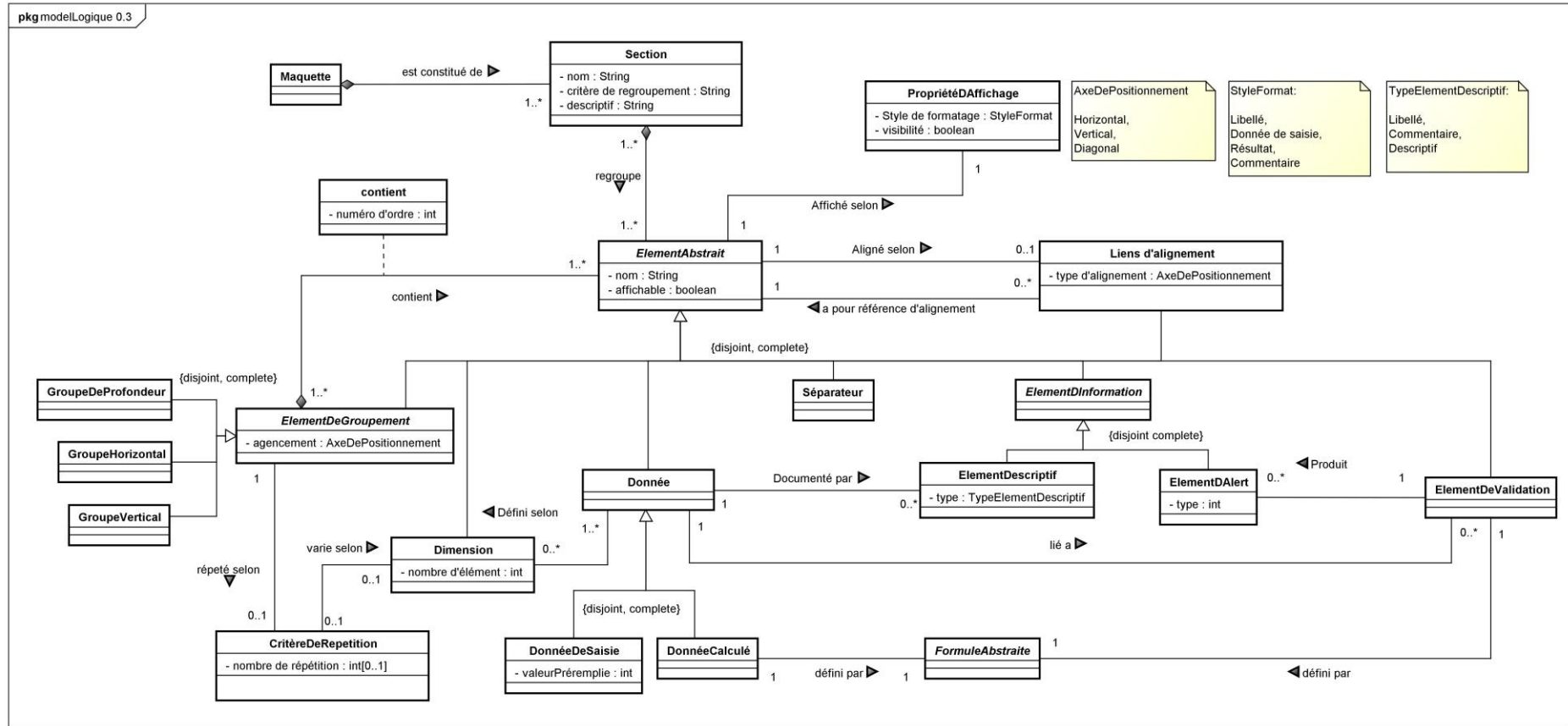
Maintenant que nous avons parcouru l'ensemble des éléments abstraits régissant la disposition des éléments dans les sections, attardons-nous aux éléments qui permettent l'interaction avec les utilisateurs. Ces éléments sont différenciés selon les sous-types suivants :

- Les **éléments descriptifs** : ils représentent des **éléments d'information** utilisés dans le but de diriger l'utilisateur. On retrouve ainsi différents types d'éléments descriptifs allant du libellé au commentaire annoté en passant par le descriptif complet au format texte.
- Les **données** : il s'agit des éléments avec lesquels les utilisateurs vont interagir soit pour fournir de l'information (les Données de saisie) soit pour connaître les résultats calculés par le tableur (les données calculées).
- Les **éléments d'alerte** : ces **éléments d'information** permettent de définir de quelle manière et avec quel message les utilisateurs doivent être avertis en cas d'introduction d'une valeur erronée, d'anomalie dans le calcul des résultats ou plus généralement lorsqu'une action réalisée par l'utilisateur ou le système requiert une attention particulière de l'utilisateur.

Du côté des traitements qui devront être réalisés par le tableur, nous retrouvons la notion de formule.

6.2 Vue d'ensemble

Le schéma suivant représente un vue global de l'ensemble de éléments impliqués dans la construction du modèle physique.



6.3 Règles de traduction

Dans cette section nous regroupons l'ensemble des règles de traduction qui, partant du modèle de calcul conceptuel, permettent l'obtention d'une feuille de calcul qui permet l'instanciation du modèle de calcul.

Nous proposons ici une traduction réalisée en deux phases.

La première permet de traduire les concepts du modèle conceptuel en éléments abstraits mais néanmoins plus proches des concepts fondamentaux d'un tableur. A la fin de cette première phase, nous disposons d'un modèle de calcul logique dont l'objectif est d'être traduisible dans n'importe quel tableur. Dans le cadre de ce mémoire, nous nous limitons néanmoins à la généralité des tableurs Microsoft® Excel et Libre Office.

La deuxième phase consiste en la traduction du modèle logique, obtenu lors de la première phase, en feuilles de calcul d'un tableur cible tels que ceux cités précédemment.

6.3.1 Du modèle conceptuel au modèle logique

Lors de la traduction des éléments du modèle conceptuel en élément logique, l'utilisateur doit poser certains choix qui auront une influence sur la disposition des différents éléments de la feuille de calcul final.

Un modèle de calcul correspond à une maquette. Il n'y a pas de réelle traduction dans ce cas, juste une correspondance un à un entre le concept du modèle conceptuel et le modèle physique.

Éléments modèle conceptuel	Éléments modèle logique	Règle de transformation
Modèle de calcul	Maquette	Définition d'une nouvelle maquette

Partant des différentes grandeurs du modèle et du contexte du problème de calcul, on dégage un ensemble de groupes permettant de regrouper les grandeurs par intérêt commun. Chacun de ces groupes forme ainsi les sections de la maquette et contiendra uniquement les grandeurs partageant le même critère de regroupement.

Éléments modèle conceptuel	Éléments modèle logique	Règle de transformation
Groupe de grandeurs partageant un même intérêt/critère de regroupement	Section	A chaque critère de regroupement correspond la définition d'une nouvelle section

Éléments modèle conceptuel	Éléments modèle logique	Règle de transformation
Nom d'une grandeur	ElementDescriptif	Type Libellé
Description d'une grandeur	ElementDescriptif	
Unité de mesure d'une grandeur	ElementDescriptif	
Pertinence d'une grandeur	Style de formatage	
Accessibilité d'une grandeur	Style de formatage	
Domaine de valeur (intervalle/énumération)	ElementDeControl + ElementD'alerte	
Domaine de valeur(réel)		
Grandeur simple	ElementDeCalcul	
Grandeur dimensionnée	ElementDeCalcul contenu dans un à plusieurs	

	ElementDeGroupelement répété suivant les dimensions de la grandeur	
Dimension	Dimension	
Donnée constante	ElementDeCalcul avec contenu	
Donnée utilisateur	ElementDeCalcul sans contenu	
Grandeur calculée	ElementDeCalcul avec contenu	
Contrainte	ElementDeControl + ElementD'alerte	

6.3.2 Du modèle logique au modèle physique

Une maquette correspond à la création d'un nouveau classeur. Il n'y a pas de réelle traduction dans ce cas, juste une correspondance un à un entre le concept du modèle logique et le modèle physique.

Éléments modèle logique	Éléments modèle physique	Règle de transformation
Maquette	Classeur	Création d'un nouveau classeur

Une section n'a pas de correspondance exacte avec l'un des concepts du modèle physique. Dans ce cas, une section correspond plutôt à une contrainte relative à la disposition des différents éléments dans le classeur. On veillera à séparer distinctement les éléments faisant partie de sections différentes. Par exemple si le modèle logique définit deux sections contenant des éléments abstraits disposés uniquement sur un axe horizontal, on définira chacune des sections comme des feuilles de calcul différentes.

Éléments modèle logique	Éléments modèle physique	Règle de transformation
Section	Plages de cellules et/ou feuilles de calculs distincts	Pour chaque section, au choix : <ul style="list-style-type: none"> création d'une nouvelle feuille de calcul création d'une nouvelle plage de cellule

Conclusion

L'objet de ce mémoire était d'apporter une aide aux utilisateurs dans la construction de solutions à des problèmes de calcul basés sur des tableurs, et ceci en utilisant des méthodes simples, accessibles au grand public et réduisant les risques d'erreur.

Il existe en effet des solutions relativement simples et accessibles du grand public pour la gestion de la visualisation des données multidimensionnelles. Il s'agit principalement du concept de tableau-pivot dans Excel qui offre déjà de multiples possibilités (celles-ci n'ont pas été prises en compte dans notre étude). Mais ce concept est souvent mal utilisé par les utilisateurs et le problème de départ est souvent mal compris ou mal décrit. Et ceci peut avoir de graves conséquences dans la validité des résultats calculés par le tableur.

Nous avons dès lors axé notre approche sur la formalisation du problème en concepts gérables par un tableur. Notre solution consiste en deux phases de traduction du problème de calcul qui permettent de l'implanter physiquement dans un tableur. La première phase est la transformation du modèle conceptuel en modèle logique qui est ensuite lui-même traduit dans la deuxième phase en modèle physique.

Notre approche présente cependant certaines limites. La limite majeure étant qu'elle ne gère pas les sous-modèles et les calculs récursifs, qui sont pourtant présents dans le modèle de calcul de base de (Hainaut, 2005). Deuxièmement, les traductions sont gérées pour un seul modèle, celui de (Hainaut, 2005), qui ne gère pas explicitement la hiérarchie entre dimensions. Ensuite, bien que les règles de traduction soient utilisables pour la plupart des tableurs, elles ont été pensées sur base des contraintes techniques du tableur Excel, qui est cependant la solution la plus utilisée, surtout par le grand public et présente les mêmes concepts de base que la majorité des tableurs. Enfin, nous avons dans ce cas seulement une méthode de traduction, une suite de règles à suivre, mais il existe également des solutions logicielles au même type de problèmes ainsi que certaines solutions proposant l'optimisation des formules utilisées dans les cellules. Ces solutions ne sont cependant pas accessibles au grand public qui fait l'objet de cette étude.

Bien que basique, notre approche apporte néanmoins une vue d'ensemble des différents concepts intervenant dans les traductions grâce à l'introduction des méta-modèles. De plus, nous avons tenté une nouvelle approche en introduisant le concept du modèle logique dans cette méthode de traduction. Ceci permet, lors de la phase de traduction, de s'intéresser plus précisément à l'utilisation des feuilles de calcul par les utilisateurs.

Afin de pousser plus loin cette recherche dans le futur, il serait intéressant selon nous de trouver un moyen de combiner les fonctionnalités avancées proposées par Excel comme les tables-pivot aux règles de traduction proposées dans notre étude. Il serait également intéressant d'étudier les aspects de modularité et de sous-modèles par rapport à la gestion multi-classeurs.

Bibliographie

ClassSheet - model-based, object-oriented design of spreadsheet applications. **Fabian, Christ, et al.** **2007.** Tools Europe 2007, Zurich : ETH Zurich, Chair of Software Engineering, October 2007, Vol. Vol 6.

Elzbieta, Malinowski et Esteban, Zimányi. **2008.** *Advanced Data Warehouse Design From Conventional to Spatial and Temporal Applications.* Berlin : Springer, 2008. ISBN: 978-3-540-74404-7.

Gabor, Maksay et Yves, Pigneur. **2008.** *Modéliser par l'exemple pratique des tableurs et des bases de données.* Lausanne : Presse polytechniques et universitaires romandes, 2008. ISBN: 978-2-88074-792-3.

Gris, Myriam. **Février 2013.** *Microsoft® Office 2013 : Word, Excel, PowerPoint et Outlook.* St Herblain : ENI, Février 2013. Vol. Microsoft® Excel 2013. ISBN: 978-2-7460-7896-3.

Hainaut, Jean-Luc. **2005.** *Bases de données et modèles de calcul Outils et méthodes pour l'utilisateur Cours et exercices corrigés (4e édition).* Paris : Dunod, 2005.

Leclercq, JP. **2008.** Cours de Théorie des Graphes et réseaux de Petri. Namur : UNamur - Faculté de Science Informatique, 2008, 1, p. 13.

Lilen, Henri. **2013.** *LibreOffice pour les nuls.* Paris : First interactive, 2013. ISBN: 978-2-7540-4920-7.

Morley, Chantal. **2012.** *Management d'un projet système d'information – 7e édition.* Paris : Dunod, 2012. pp. 139-148,253-283.

Petit, Michaël. **2009-2010.** Systèmes d'information I : introduction et modélisation (ECGE-B110). Namur : UNamur - Faculté de Sciences Economiques, Sociales et de Gestion, 2009-2010, 4. Problèmes de calcul.

Ramdoyal, Ravi. **2012-2013.** Intéractions Homme-machine (IHDCM030). Namur : UNamur - Faculté d'Informatique, 2012-2013, 2. La conception de systèmes interactifs, pp. 169-189.

Annexes

Voir dossier ***annexes***